DESIGN AND DEVELOPMENT OF A MASTER-SLAVE TELEOPERATED ROBOT

by

GIRI N. P.

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MAY 1990

DESIGN AND DEVELOPMENT OF A MASTER-SLAVE TELEOPERATED ROBOT

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

GIRI N. P.

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MAY 1990

CERTIFICATE

This is to certify that the present work DESIGN AND DEVELOPMENT OF A MASTER-SLAVE TELEOPERATED ROBOT by Sri. Giri.N.P has been carried out under our supervision and that it has not been submitted elsewhere for a degree.

A Ghosh

Prof. A.Ghosh
Head, Mechanical Engineering
IIT, Kanpur

K Sri Ram

Prof. K.Sriram
Head, Nuclear Engineering
IIT, Kanpur

TO

MY PARENTS

ABSTRACT

Name of student : Giri N.P.   Roll No. 8820503

Degree for which submitted : M.Tech : Dept: Mech. Engg.

Thesis title : "Design and Development of a master slave teleoperated robot"

Names of thesis supervisors : 1. Dr. Amitabha Ghosh, ME
2. Dr. K. Sriram, ME/NET

Month & year of thesis submission : May 1990

This project involves development of a master-slave teleoperated robot which can meet some of the national requirements in teleoperator technology and can act as a stepping stone in the nascent area of ´telerobotics´.A slave robot which can execute possible motions of human arm is selected. A kinematically equivalent master arm with good hand adaptability is fabricated. Master arm joint parameters are sensed with position sensors and ADC. A control software which converts ADC output to joint parameters of the slave and executes interpolated motion between sampled points was developed. The overall system was realised and tests were conducted to evaluate the performance. Finally suggestions for future extension programmes are made.

ACKNOWLEDGEMENT

I extend my sincere gratitude to my thesis supervisors Prof. Amitabha Ghosh and Prof. K. Sriram for giving me complete freedom, constant encouragement, valuable guidance in mechanical and control areas, and parental care and affection. I am indebted to Dr. A.Ghosh also for bringing me into the area of robotics and for the all round advice he has given during my programme. I owe to Dr. Sriram for clearing my control concepts and giving timely suggestions.

I am grateful to Prof. R.N. Biswas, EE who helped me in clearing my views on teleoperator control and building a foundation in digital electronics through his course.

I wish to thank Mr. R.M. Jha of manufacturing sci. lab whose constant cooperation has helped me in realizing the master arm in time, inspite of adverse factors that came in between. I also thank Mr. Prem Praksh Gupta ,Mr. Bharathia and other staff in manufacturing sci. lab for various help renderd during fabrication.

I wish to express my gratitude to Dr. S.R. Pandian, RE for greatly helping me at various levels of this project. Mr. A. Bagchi has suggested alternatives in software solution and Mr.S.Sen has helped in solving

hardware problems.I wish to thank them. I am grateful to all personnel especially Mr. Samiran Mandal,Miss. Sushma sinha & Mr. Laxmi prasad in Robotics lab for giving me constant cooperation.

My parents constantly inspired me, and gave me a peaceful time here, managing everything at home themselves in my absence. I am deeply indebted to them.

I am grateful to Mr. M.K. Abdul Majeed, Head and Mr. George Koshy, of Aerospace Mechanisms Division, Vikram Sarabhai Space Centre for encouraging me in higher studies and giving me this opportunity.

I am indebted to Mr.Vivek Shukla, Robotics lab and Mr.B.L. Arora, Design & Mfg. Cell for their help in presenting this thesis.

CONTENTS

LIST OF FIGURES /CHARTS

FIGURES

CHARTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Teleoperation is the extension of a person´s manipulative, perceptual and cognitive skills to a remote location. The increased need for remote manipulation in environments hazardous to human health or survival, like nuclear, undersea, outer space, etc has been widely recognized. Many tasks in such environments are complex, unpredictable and unplanned and therefore cannot be done by autonomous machines like robots. Human perception, planning and control are required to perform these tasks successfully and safely. A teleoperator couples human decision making and motor control functions with a manipulator working in a remote/hostile environment.

Teleoperator includes artificial sensors, arms and hands, a vehicle for carrying these and communication channels to and from the operator

The manipulator arm and vehicle will be in hostile environment and the operator will be in clean environment. The manipulator performs its function under direct control of the human operator. Fig.1 shows the principle of teleoperation. The operator moves the hand controller by viewing the image of hostile environment & robot on the

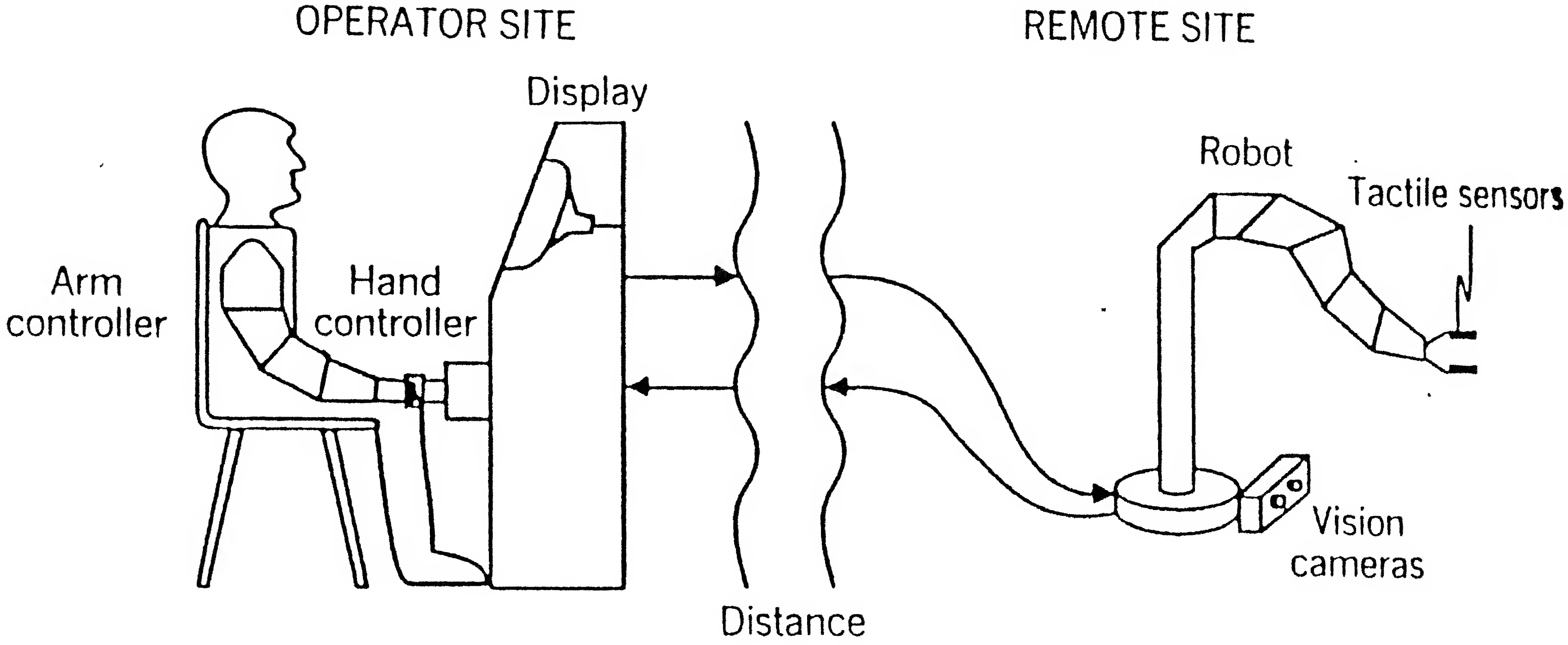


Fig 1. PRINCIPLE OF TELEOPERATION

(Courtesy [5])

display. Feed back from visual sensors and tactile sensors on the robot gives him a feel that he is directly present in remote/hostile site and thus helps in easy manipulation.

1.2 CONTROL OF TELEOPERATORS

Traditionally, the aim of teleoperation has been to give the user charge of the movements of remote manipulator in an ergonomically satisfactory way. This is done in three ways:

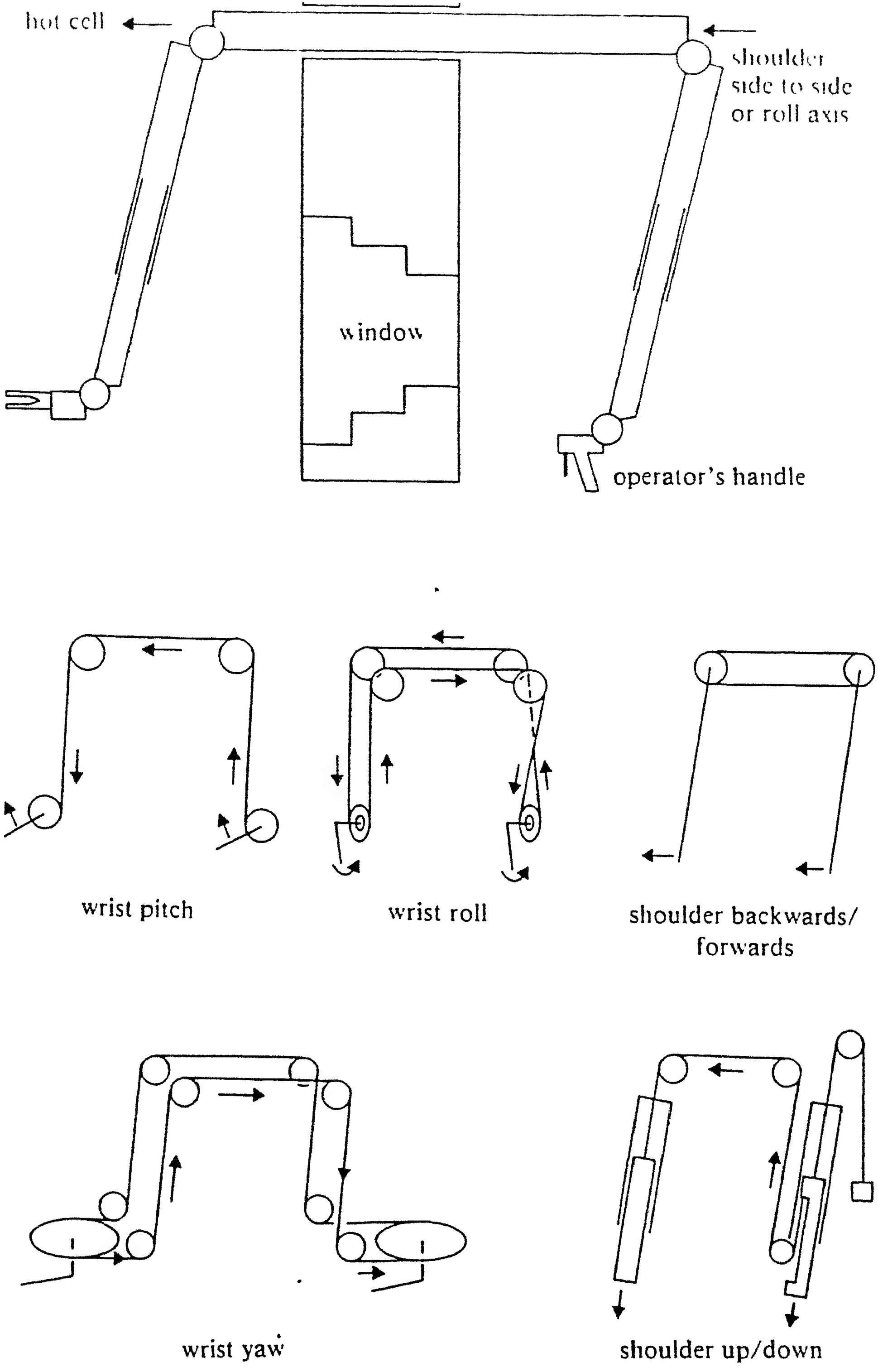
1) mechanical master-slave telemanipulators
2) powered telemanipulators with unilateral control
3) Powered telemanipulators with bilateral control

1.2.1 Mechanical Master Slave Manipulators

In these, the motions of the operator´s hand are reproduced by mechanical linkages connecting a slave arm with a gripper to a master arm ending in hand control. The mechanical connection allows the user to feel the forces on the slave and so there is some feedback. Fig.2 shows the principle of cable transmission for 5 dof of such manipulator, the sixth being produced by rotating the whole system around a sleeve through the wall.

1.2.2 Unilateral Control

In unilateral control, there is no force feedback.



wrist pitch

wrist roll

shoulder backwards/ forwards

wrist yaw

shoulder up/down

Fig 2. Cable transmission for 5 dof of mechanical master- slave manipulator.

(Courtesy [15])

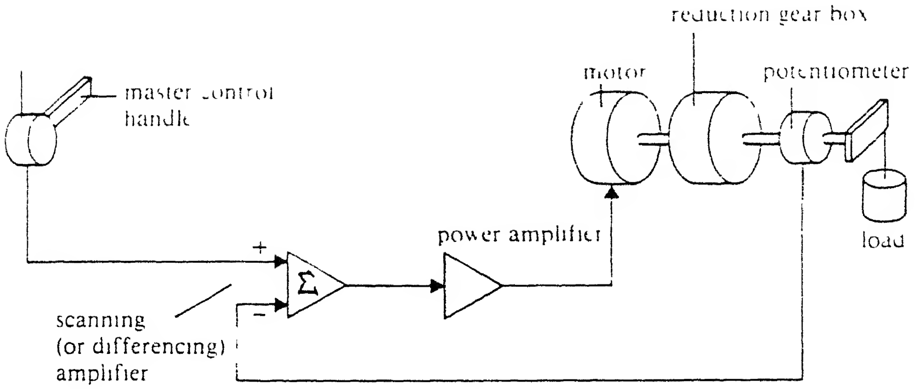


Fig 3. Servo for Unilateral control

(Courtesy [15])

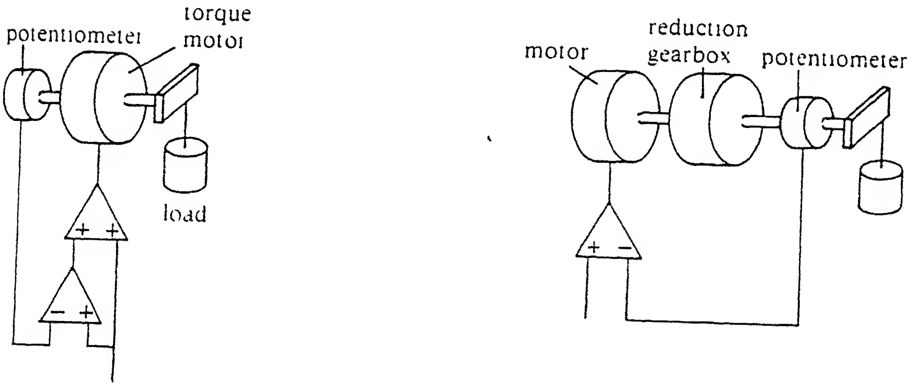


Fig 4a. Position servo for Bilateral control

(Courtesy [15])

The operator´s control device may take the form of a master arm whose joint angles are measured and drive the corresponding joint servos in the slave arm. They are used wherever mobility is required. A servo for unilateral control of one joint is shown in Fig.3. It is open-loop in that there is no feedback from the slave to the master, but there is a position feedback within the slave-half of the system. In this loop, if the position signal from the slave is equal to that from master, there is no motor movement.

1.2.3 Bilateral servo control

Here the force on the slave arm is fedback to the master arm for the operator to feel. Force reflection helps the operator to consider himself present in the remote location and thus induces ´telepresence´. Systems with telepresence are symmetrical where master can also be moved like slave just as in the case of mechanical master-slave manipulators.

There are many types of bilateral control systems. The master can be designed to take position or force as its input, the controlled variable at the slave also being position or force and similarly for the return loop. Actuators are provided at the master end as well to enable

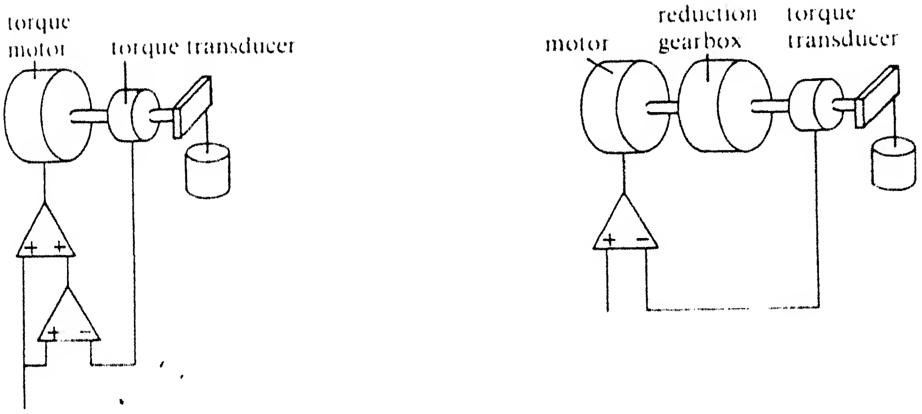


Fig 4b. Force servo for Bilateral control

(Courtesy [15])

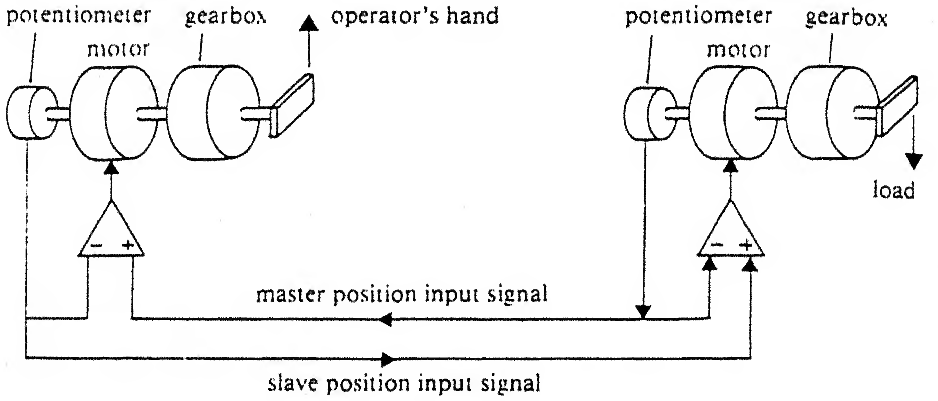


Fig 5. Position- Position bilateral servo control system

(Courtesy [15])

it to be back driven and torque motors generally act as transducers of torque.

At the master and slave ends, position or force servos can be used. A position servo (Fig.4a) maintains the output shaft angle equal to the input commanded angle regardless of load. A force servo (Fig.4b) maintains the output torque proportional to the commanded torque regardless of position. Some torque opposes the torque produced by the motor so as to stop the shaft moving. When this is true the signal from torque transducers balances the input signal and the shaft does not turn. The summing amplifier is required, since the motor at master end is back-drivable also and so even if there is no error a minimum torque must be kept. In other cases types, gears can support the load and motor current can be zero.

There are different types bilateral control systems [15] [19]. Fig.5 shows position-position bilateral servo control system. When master arm is moved, the slave follows and the operator experiences the forces on the slave, though they are not explicitly fedback.

With these basic control systems teleoperators have evolved from mechanical master-slave systems to telerobots.

1.3 EVOLUTION OF TELEOPERATORS

Teleoperators in the primitive form like tools,weapons and cranes which extend human capabilities were being used since eighteenth century. Modern teleoperators were developed in the 1940's for radioactive material handling in "hot cells". The first teleoperator developed by Goertz at Argonne National Laboratory was a mechanical master slave manipulator using pantograph mechanisms. Electrical servomechanisms soon replaced mechanical linkages. Shortly, television helped the operators located far away to view remote site. In 1960's force reflection, two-arm teleoperators and head mounted displays produced remarkable visual 'telepresence'. Soon teleoperators with video cameras were attached to submarines opening up deep sea applications. The Sixties also saw the first space manipulator, when Surveyor III landed on moon in 1967 and collected soil samples. By 1970, industrial robotics was coming into full development with computer vision, force sensing etc. Chart 1 shows the evolution of teleoperators.

Current developments are mainly in telerobotics where supervisory control is exercised by a man over a slave robot.

1.3.1 Telerobotics

Telerobotics is a relatively new area of teleoperator control, made possible by intermingling teleoperation with robotics. It is analogous to a supervisor monitoring & directing the activities of a subordinate. Here the subordinate will be a robot with its own computer.

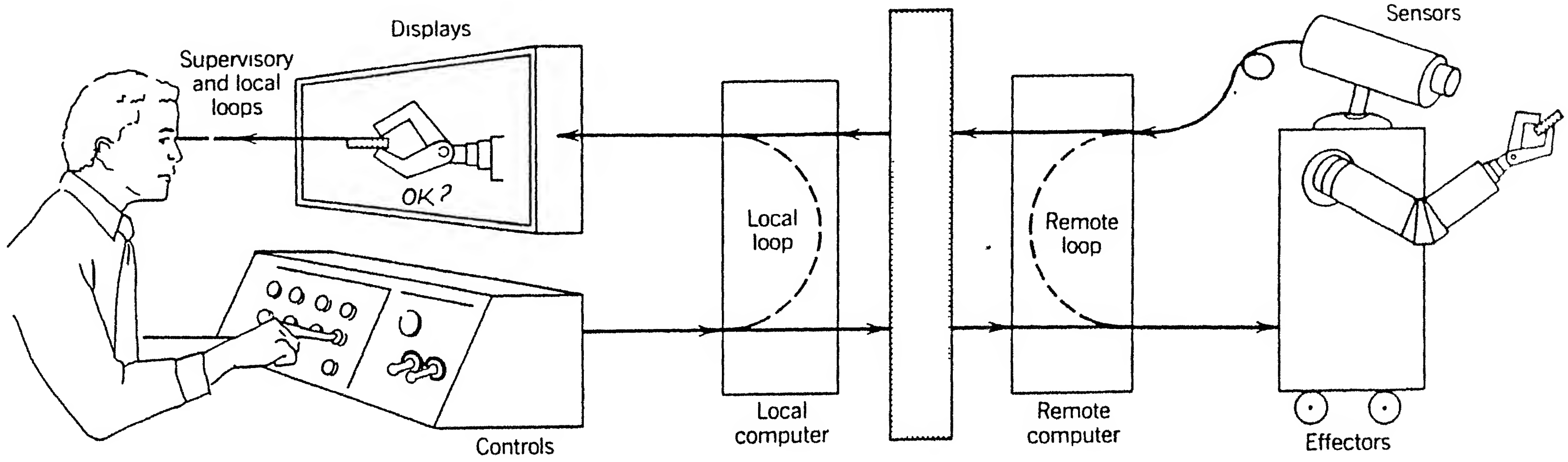


Fig 6. CONCEPT OF TELEROBOTICS

(Courtesy [3])

Supervisory control, though introduced in the 60´s [3] to reduce delay in teleoperation, has found application only the 80´s for advanced teleoperation. It is applied in ´telerobotics´ in which the human operator ´intermittently communicates to a computer information about goals, constraints, plans, suggestions and orders relative to a limited task, getting back information about accomplishments, difficulties, concerns and, as requested, raw sensory data - while the subordinate telerobot executes the task based on the information received from the human operator plus its own artificial sensing and intelligence´. [10]

The human operator supplies largely symbolic commands to the computer. However, a fraction of these commands must still be analogic (hand control movements etc.) in order to point to objects or to demonstrate to the computer relationships that are difficult for the operator to put into symbols. Fig. 6 shows the concept of telerobotics.

The local or human interactive computer thus must be human friendly, able to indicate that it understands the message or to point out that a specification is incomplete and if so help the operator to edit the message correctly. It also must interpret signals from the distant telerobot, storing and processing them to generate meaningful graphic displays. The local computer should have a knowledge base and a model of the teleoperator and task environment. The subordinate remote computer which accompanies the tele robot, meanwhile must receive commands, translate them and perform execution closing each control loop through the appropriate telerobot actuators and sensors.

CHART 1

Year	Event	Inventor/agency
1945	Mechanical master-slave manipulator	Goertz (Argonne National Lab, Chicago).
1954	Electrical servo manipulator	Goertz, Thompson
1954	manipulator mounted on mobile vehicle	Argonne National Lab.
1958	10 dof electro hydraulic arm with force reflection	Mosher (General Electric)
1961	Heavy manipulators with large capacity (300 kg to 1 ton)	Jean Vertut
1963	Underwater teleoperation Cable controlled underwater research vehicle.	U.S. Navy
1967	Concept of supervisory control	Ferrel, Sheridan
1967	Surveyor III, Lunakhod land in moon and collect samples	USA/USSR
1969	Idea of computer-aided teleoperation	Draper Laboratory MIT
1976	supervisory controlled Viking spacecraft with arms to collect soil samples from mars.	NASA
1970´s	Industrial robotics	Unimation, Hitachi General Electric
Present	Telerobotics, teleactuation, teletouch, multiperson control, supervisory control	Various institutions

1.4 CURRENT APPLICATIONS OF TELEOPERATORS

1.4.1 Space

Teleoperation have been used in space exploration for driving vehicle (lunakhod) on the moon and collecting soil samples from planetary surface. (Surveyor on the moon and Viking on mars). Operation of manipulators on planetary surfaces presents the problem of time delay if they are controlled from earth. So, it becomes necessary to either work slowly or introduce some degree of automatic control. Viking arm which sampled soil on mars has been programmed to dig and collect samples when moved to a location by terrestrial control.

Current example of space teleoperator is the 20m long remote manipulator system (RMS) [10] carried aboard the US space shuttle. It has 6 dof and is controlled directly by a human operator viewing through a window or over a video and using two, three-axis variable rate command joysticks, one for three translations and the other for three rotations,. Since operations are performed in zero gravity, the arm can be light: It can not support its own weight on earth. In space, with low angular rates, it can deploy and retrieve satellites of hundred´s of kilograms with ease. Master slave control is not appropriate at low angular rates. So some degree of autonomy is imparted.

US space program is focussing on the design of a ´flight telerobotic servicer´ (FTS) , a general purpose manipulation device. It is to have two arm of size of human arms. Eventually FTS would be attached to a free flying thrust platform and also be provided a modicum of computer intelligence beyond simple control laws.

1.4.2 **Nuclear power**

Much of the development of teleoperator has been stimulated by the need to handle radio active materials Nuclear ´hot´ laboratories continues to use the maximum number of teleoperators. Most extensive & successful of such systems are MA - 23 type designed by Vertut and Master-slave type designed by Goertz. They use mechanical master slave system with transmission passing through the wall separating master´s clean room from ´hot cell´. Operator watches slave operations through a radiation opaque window or a video. They are widely used in radio chemistry or isotope applications, spent fuel handling and reprocessing. They had to be designed to remain inside radio active nuclear fuel processing enclosures for many months and thus be extremely reliable. Nuclear teleoperators are now becoming sufficiently mobile and dexterous for doing tube repair work on steam boilers, plant maintanence and, decommissioning and decontamination of reactors. Recently, nuclear engineered advanced telerobots (NEATER) are introduced in some nuclear installations in UK [16]. They have converted PUMA 762 produced by Unimation Ltd. into a telerobot, after improving seals, circuitry etc to prevent irradiation effects and plan tc use them for decommissioning & nuclear waste management. (also see 1.5)

1.4.3 Deep sea

Offshore oil and gas industries use remotely operated submersible vehicles in well head completion operations, monitoring of pipe lines and inspection of welds. These applications were initiated by cost and risk factors involved in such operations. Most undersea manipulators are hydraulic to withstand high forces. They are usually tethered to a mother ship and are manned or unmanned. Remotely operated vehicles are also employed to aid scientific investigations by marine biologists and geologists, and for mineral module collection from sea bed. Argo and Jason Junior are such unmanned submersibles used in 1987 expedition to the ´titanic´ [10].

1.4.4 Medical

Medical field employs many arm prostheses which include digital processing of myoelectric signals and modern mechanical actuators, augmented by natural looking hands. Teleoperated electro mechanical guide dogs have been developed to guide the blind.

One form of teloperator used in surgery is the endoscope [6], a coherent fibre optic bundle with tubes for conveying fluids, a gripper at the end for minor gripping and surgery.

Teleoperators are also widely used in terrestrial mining, construction and maintanenece, firefighting, surveillance and military operations.

1.5 INDIAN SCENARIO

Indian nuclear research programme have been using various remotely operated′ mechanisms′ such as fuel handling machine for on line fuelling of reactors, retrieval system of irradiated fuel clusters, decladding machines, device for handling fuel, isotopes and guidetube assemblies, remote cutting facilities etc. Manipulators employed in nuclear sector have been of mechanical master-slave type which are descendants of Goertz machines. Recently, work has begun on the use of electric servomanpulators and robots for fuel decladding and spent fuel reprocessing. Some robots are employed in online sampling of processing. Ramakumar & associates describe various remote handling systems used in nuclear installations in [1].

The trend in nuclear sector is towards teleoperated mobile robots which will enhance applications in current nuclear research and decommissioning of spent rectors.

Space research also plans to use teleoperators in its ground based systems [1] . These include teleoperated arms for assembling and testing explosive devices, mobile units with teleoperated arm for launch pad inspection and, assembly robot to build space structures.

1.6 OBJECTIVE OF PRESENT WORK

The mechanical master_slave manipulators being used in Indian nuclear sector have many disadvantages. Because of length limitations it must be in a fixed position relative to the master. This restricts the use of these manipulators to manipulation in fixed installations. In future, requirements of mobile electric servomanipulators will be more to meet wide ranging applications in nuclear research and nuclear power plants, and in decommissioning of reactors like Tarapur in the near future. Some beginning has been made in electric servo manipulation, but yet in nascent stage only.

´Telerobotics´ is the state of art area in teleoperators, where major research works are being carried out abroad. In India, telerobotics research has not started yet. So to meet the national requirements in teleoperator tehnology and to establish a beginning in ´telerobotics´, the idea of master slave teleoperated robot was conceived.

1.7 SCOPE OF PRESENT WORK

The scope of present work is to develop a PC based master-slave teleoperated robot with position information fed to the slave robot by master arm controlled by an operator with less skills. It involves the design and realization of a master arm with good human interface, selection of the slave robot which has sufficient dof to execute arm controlled motions and development of a control software to achieve teleoperation. Later, mobility, telepresence by force reflection and supervisory control can be incorporated as future extension programme. The details of developing such a system is discussed in this work.

Chapter 2 covers details of master slave system requirements, slave robot and its selection criteria. The master arm design is discussed in chapter 3. Chapter 4 describes control software and electronic módules used for teleoperation. Performance of the developed system is given in chapter 5. Suggestions for future work are made in chapter 6.

CHAPTER 2

MASTER-SLAVE TELEOPERATED ROBOT SYSTEM

The system consists of a master arm, slave robot and controller, and computer. The master arm positions are supplied to the slave robot by position sensing and through the computer. The computer communicates with the robot controller regarding information from master and the controller performs robot actions using local loop. Thus it encompasses electric servc controlled teleoperation and ´telerobotics´.

2.1 REQUIREMENTS

In master-slave teleoperated systems, since man is included in the control loop, attention has to be paid in designing man machine interfaces and in selecting the slave robot. Since the objective of the present work is development of PC based master-slave system controlled by less skillful operator, use of advanced controllers [11] is not considered because of two reasons: one, operator skill requirements are high; two, they involve large computational facility for direct kinematics of master arm, inverse kinematics of slave arm and interpolations. So master arm has to be kinematically equivalent to the slave arm and both master and slave must be able to execute almost all possible motions of human arm and wrist.

With 6 dof, a manipulator can imitate human arm motions of shoulder (3 dof- roll redundant with elbow roll) elbow (2) and wrist (2) and thus locate and orient an object in space. With three major joints, shoulder and elbow movement can be duplicated. Human wrist has following limits to its three motions [4] (Fig. 7)

pitch	-90 ↔ 0 ↔ +50	=	140	elbow not extended.
yaw	-45 ↔ 0 ↔ +45	=	90	
roll	-180 ↔ 0 ↔ +90	=	270 (elbow extended)	
	-180 ↔ 0 ↔ +5	=	185 (elbow not extended)	

Another aim of the project was to mark a beginning in telerobotics where the slave has some degree of autonomy and hence must be a robot. With these requirements, the specifications of slave robot were generated.

<u>Specifications of slave robot</u>:

degrees of freedom	:	6 (wrist satisfying limits given above)
Joints	:	Electric servomotor controlled with position feed back
min reach	:	500 mm

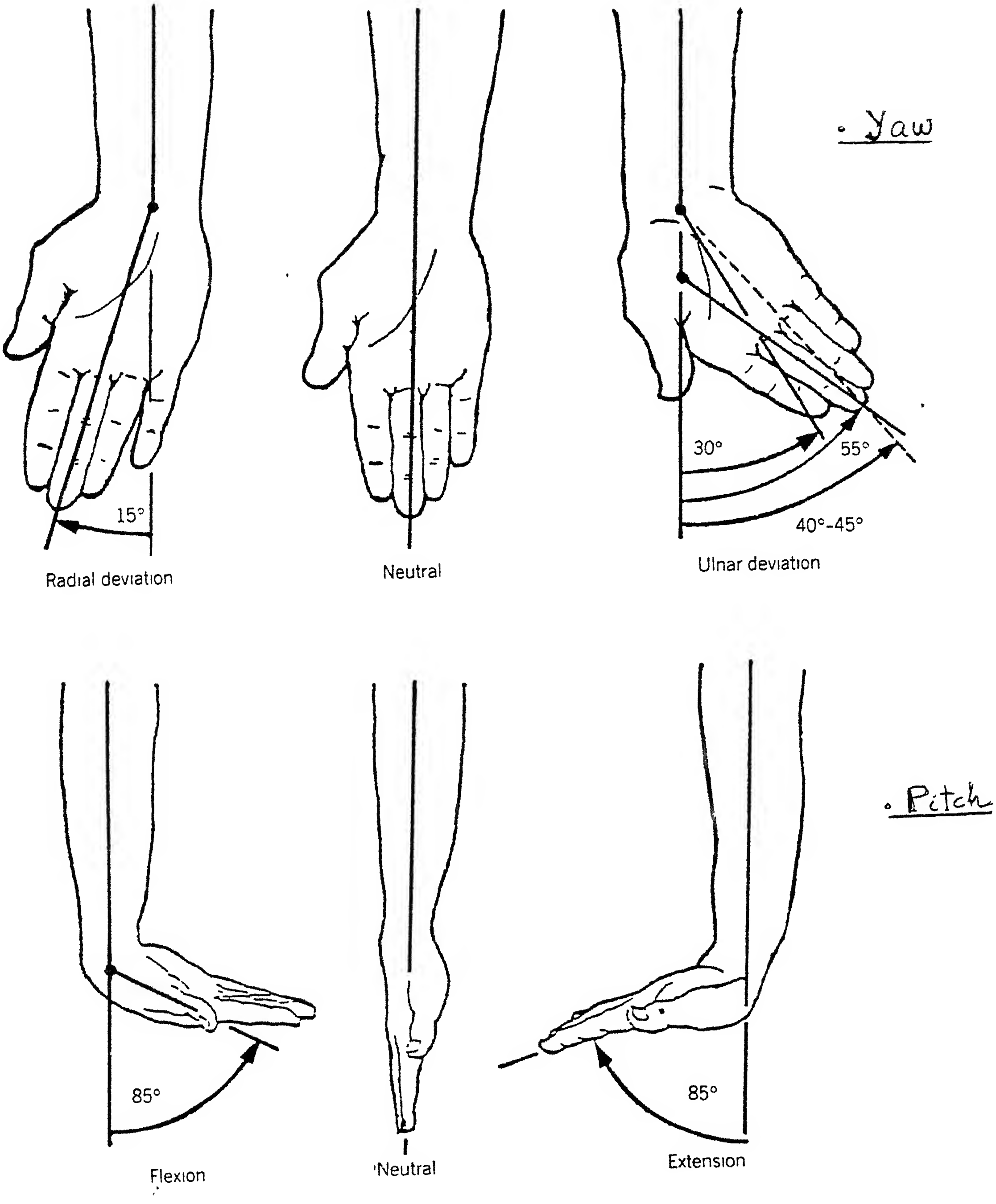


Fig 7. Human wrist motions -Pitch and Yaw

(Courtesy [5]) {elbow extended}

Payload capacity	1 kg
interfaces	PC compatible
Accuracy	± 2mm
programming options	point to point and continuous path.

Emergency stop provision

A detailed tender evaluation of various educational robots were held and MA 2000 supplied by Tec Quipment International Ltd.,UK was found to meet most of the requirements.

2.2 SLAVE ROBOT - MA 2000 [12,13]

MA 2000 consists of a robot arm, electronic controller, teach keypad and software. Robot arm has six d.c. motor powered revolute joints operating with closed loop control. It also has a pneumatic gripper. The specifications of the robot are shown in appendix 1. MA 2000 is shown in Fig. 8.

MA 2000 operates with IBM compatible host computer. The keypad is used to communicate with the computer and controller. It has additional input /output ports for controlling other devices. The robot is programmable upto 100 steps of joint position data. It has following programming options.

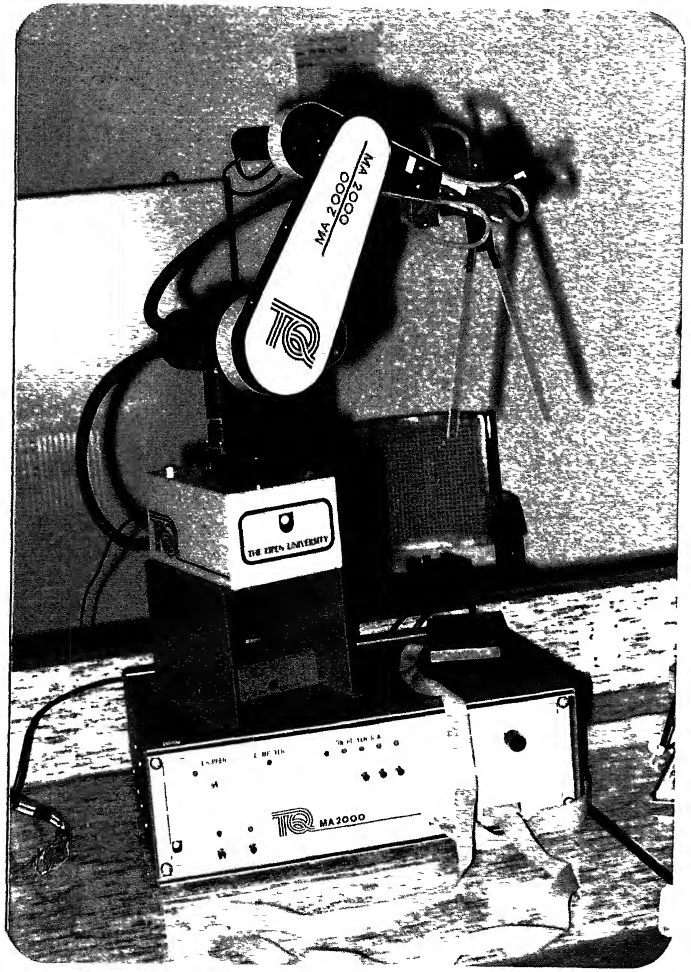
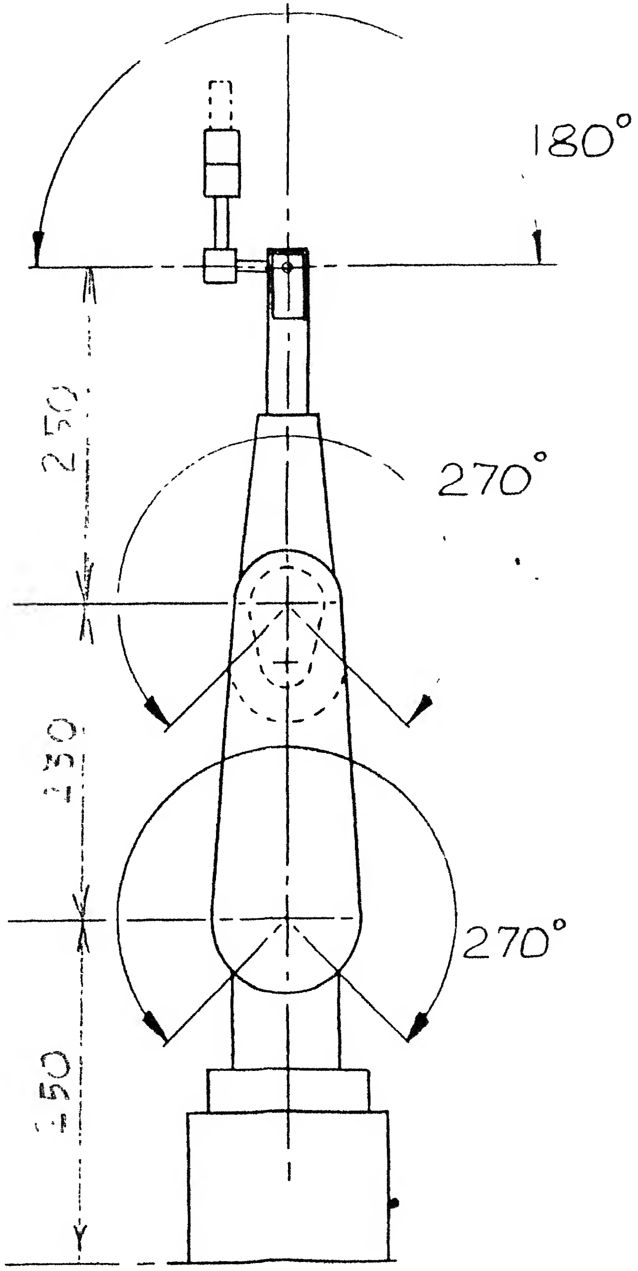


Fig 8. MA-2000 Slave robot



Other	Joints
Waist	: 270°
Yaw	: 180°
Roll	: 180°

Fig 9. MA 2000 Joint motion limits

- lead by nose - continuous path
- lead by nose - point to point
- teach keypad - point to point
- offline computer programming

In all the above options it works on teach and playback mode. Each step has information regarding step number, rate of movement (0-9 discrete), input, output, wait (time in second), jump (to a certain step), the joint parameters for waist, shoulder, elbow, pitch, yaw & ·roll and gripper status (close/open). All joints are allotted micromotion steps is the range 0-999, within their limits of motion. The limitation of joint motion is shown in Fig 9. Only one of the taught steps can be of lead by nose - continuous path mode:

2.2.1 Robot controller[13]

The controller of MA200 consists of a combined interface and motor controller board. It is designed to be driven from the host computer which supplies position information for each of the robot joints; the controller will then supply PID digital servo control of upto 6 channels to control the robot axes to achieve the required joint positions.

Motor controller uses a 6502 CPU operating at 1MHz clock rate. It´s prime task is to perform a real time PID control. It also communicates with the host computer to

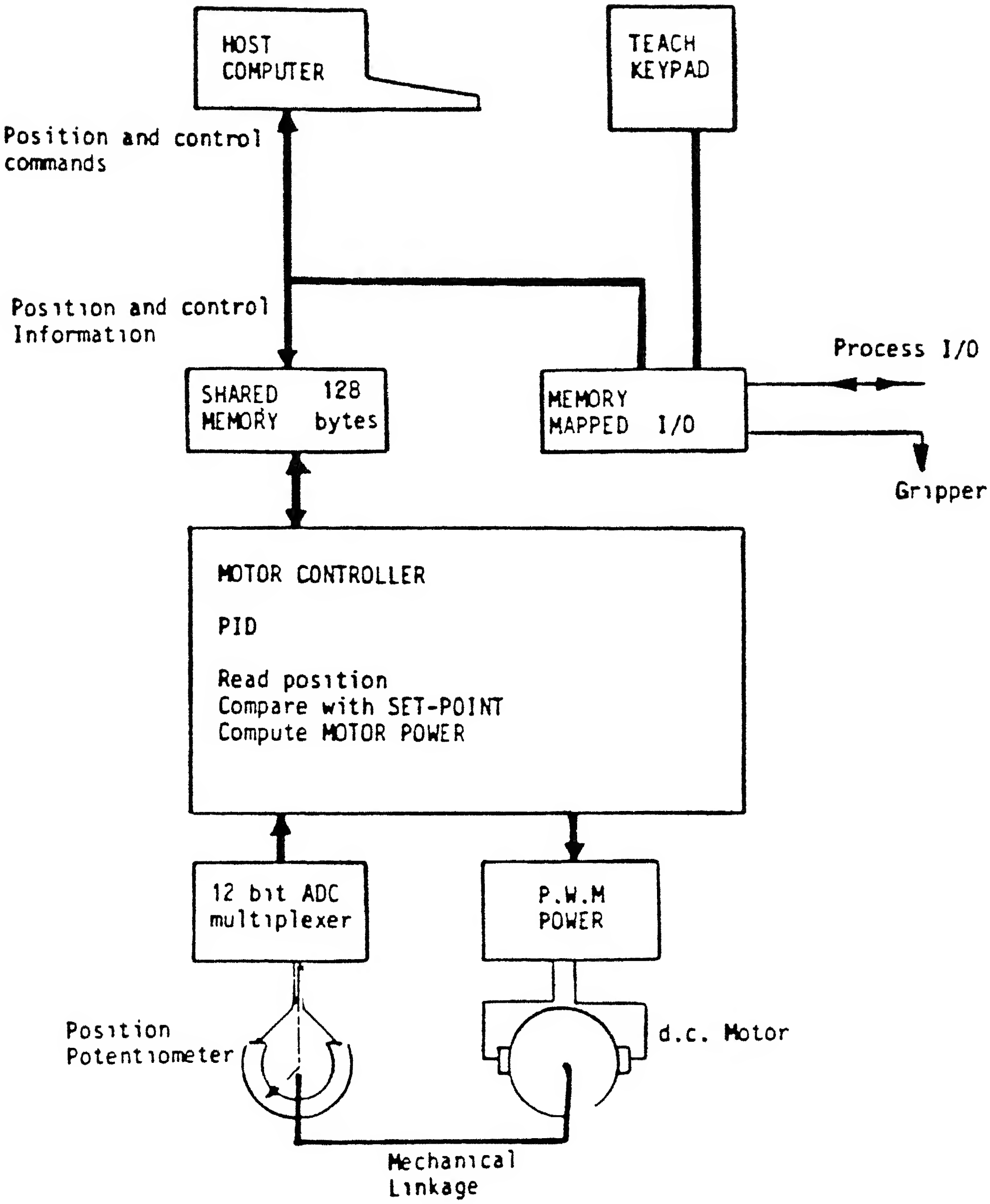


Fig 10. System configuration - MA 2000

(Courtesy [3])

obtain position information and interrogate the front panel controls. The system block diagram is shown in Fig.10.

2.2.3 The host computer

The main task of host computer is to interpret the user´s requirements and translate them to a series of motion set point commands and send them to motor controller interface. It also

- scans the teach keypad
- sets pneumatic gripper
- read status of the controller front panel
- instruct the controller interface CPU to report joint position, errors and motor power.

The operating software for the motor controller is held is ROM. Areas of RAM are used as a workspace and as a shared memory. (i.e. motor controller and computer can access this area). Four of the latches and 128 bytes of RAM (shared memory) are accessible to the host computer; the rest of the interface is not available to computer, but control information can be relayed to motor controller through shared memory. 128 bytes of shared memory are resident at addresses 0 to 7F hex. Two read only latches at addresses 80 and 81 hex and two write only latches at addresses 80 and 81 hex are used for process inputs and outputs and for keypad and front panel controls.

2.2.3 MA 2000 - IBM interface

The unit allows IBM compatible host computer to communicate with the shared memory in robot controller and with two sets of read` write latches. It uses 8255 programmable peripheral interface and associated circuitry to accomplish this. 8255 has three ports A, B, and C which can be programmed to input and output 8 bit data. The address in shared memory is allocated to port A. The data transfer to the shared memory is through port B and READ/WRITE and ENABLE signals are allocated to port C bits 1 and 0 respectively. The base address of interface set to 300 hex.

CHAPTER 3

MASTER ARM

Employing the robot in the teleoperator mode requires the design of a man machine interface with primary controls achieved through the use of hand controls. It must have 6 dof so that 6 axis slave manipulator, commanded by displacements in thę hand controller can be located at any point in workspace and be oriented at any desired attitude angle. The operator with less skill must be able to generate input commands of more than one axis with minimum or no unintended movements in other axes with one hand only. These were the prime considerations in designing the master arm. The master arm is designed kinematically equivalent to the slave arm.

3.1 DESIGN GUIDELINES

Master arm was designed to meet functional requirements of kinematic equivalence, sensor accommodation and human interaction. The wrist and grippers are designed to furnish almost all possible motions of human arm without fatigue. The arm has to be operated single handedly. Hence weight acting on arm (effective weight) influenced the design of master arm. Based on ergonomic charts[2] the effective arm weight was fixed to be less than 0.5 kg and to

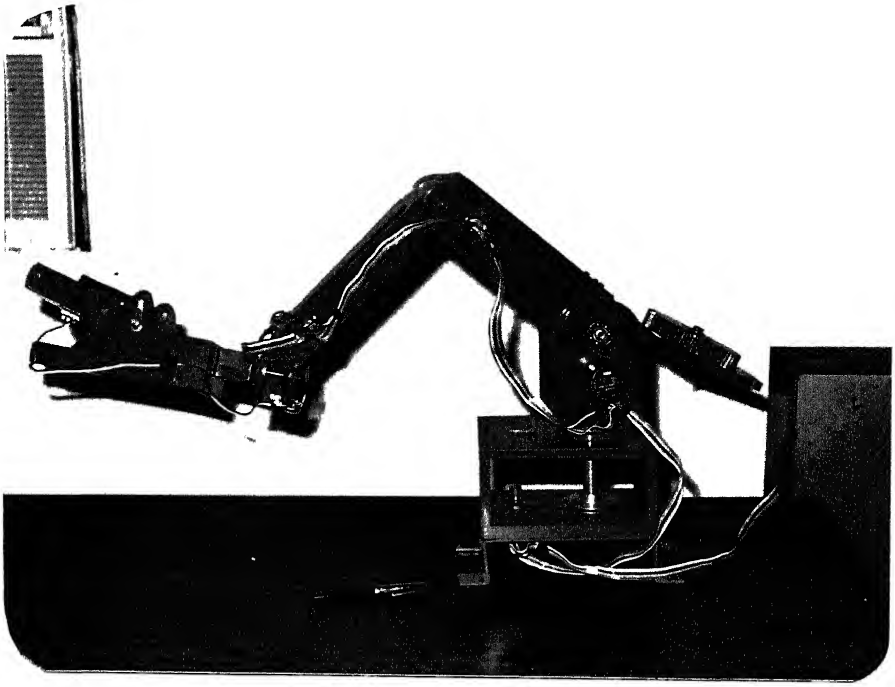


Fig 11. MASTER ARM

meet this, it was decided to use aluminium alloy as material and to add counterweight/springs at the joints to balance the weight. Wrists and gripper were designed to execute possible motion of human hand, without any interference.

3.2 LINKS AND JOINTS

All joints are provided with roller/ball bearings for smooth motion. The base link is made of cast iron block held on supports. Shoulder link is made of cast aluminium block Other links are made from 3mm thick Aluminium alloy sheet. The fabricated master arm is shown in Fig.11.

Two taper roller bearings are used for base shaft foreseeing the future need to mount the master arm on the ceiling for better adaptability to hand motions. The shaft end is extended so that shoulder link can be rigidly connected to it by grub screws. Other joints have simple ball bearings. The transmission of motion from shaft to succeeding link is by grub screws which tighten the shaft with a collar rigidly connected to the link. The joint cross sections are shown in Fig.12,13,14,15.

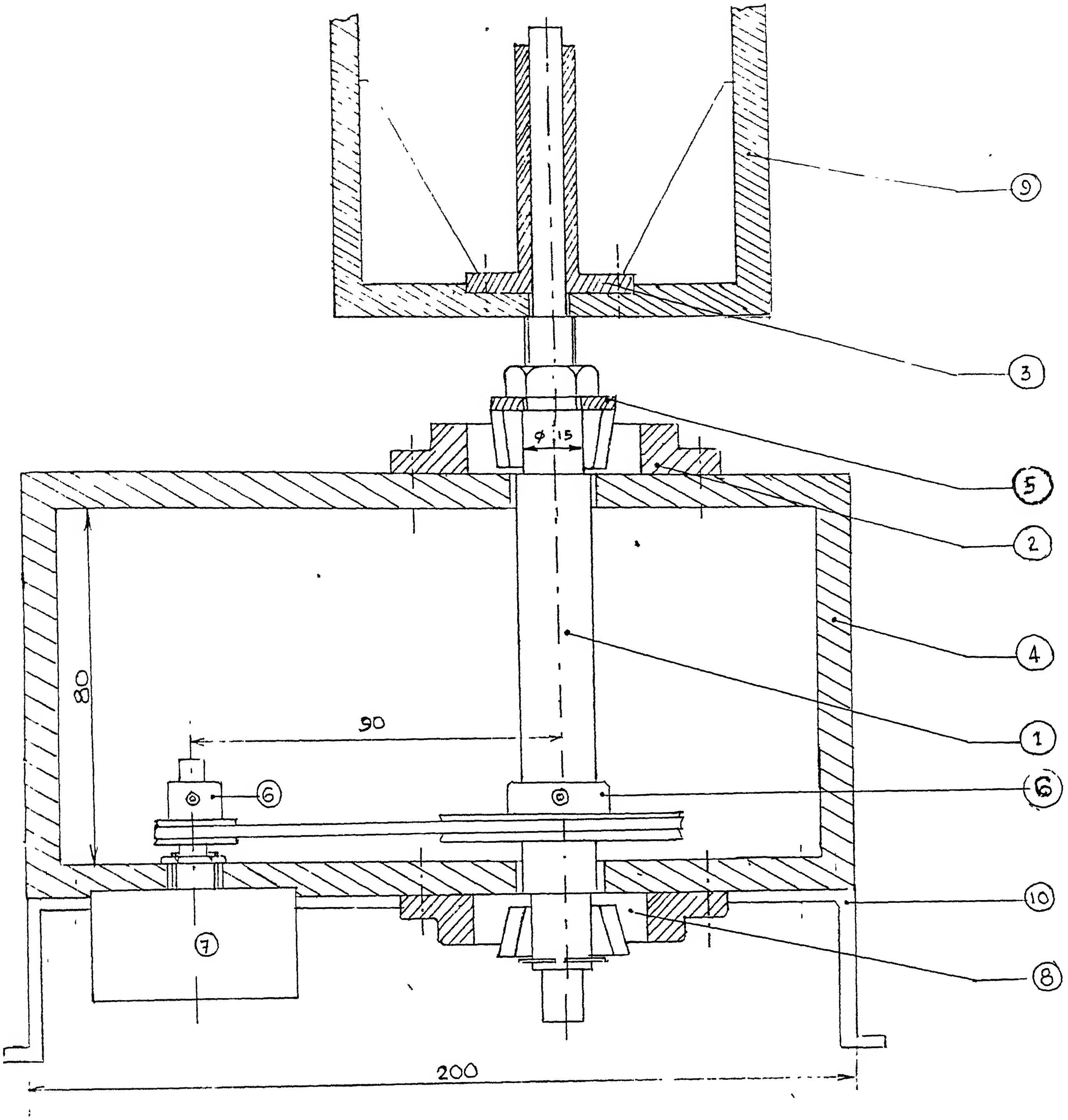


Fig.12 BASE JOINT ASSEMBLY

1. Base joint shaft
2. Bearing housing
3. Collar
4. Link Ø CI block
5. Washer
6. Pulleys
7. Potentiometer
8. Taper roller bearing
9. Shoulder (link 1) east Al block
10. Support

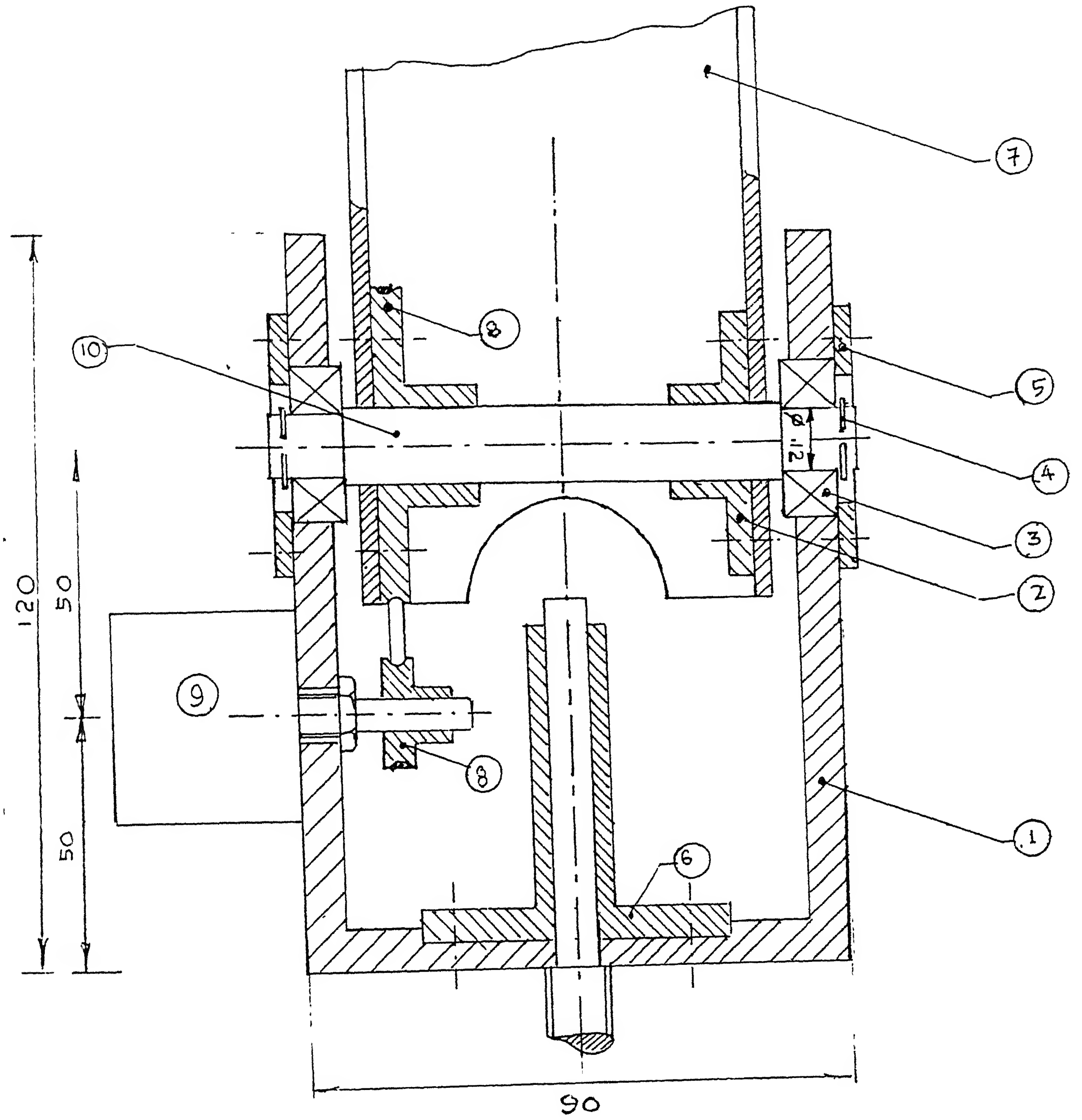


Fig.13. SHOULDER JOINT ASSEMBLY

1. Shoulder link (
2. Collar
3. Bearing
4. Circlip
5. end cover
6. Collar
7. link 2 channel Al 60 x 40 x3
8. Pulley
9. Potentiometer
10. Shoulder joint shaft

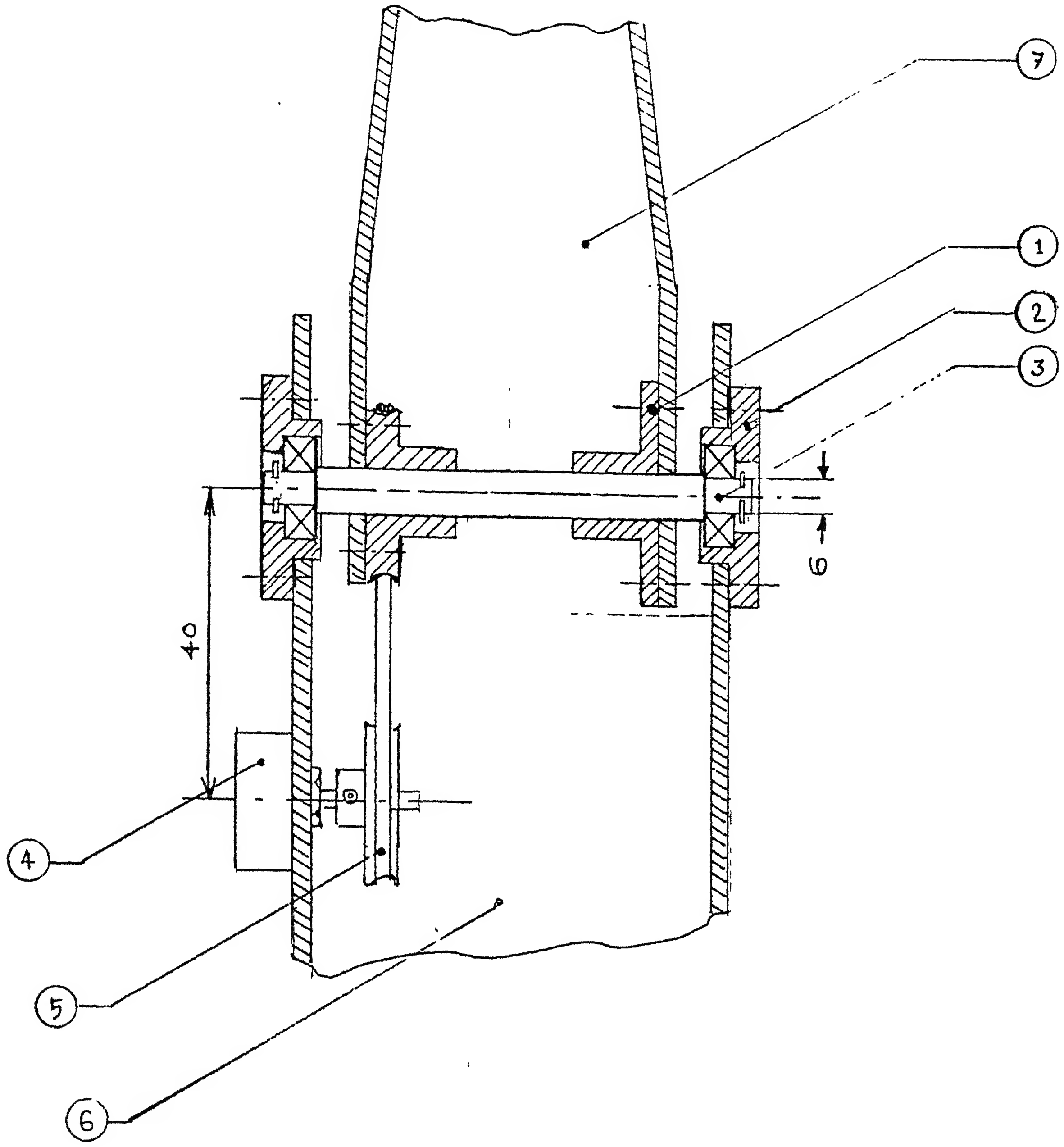


Fig.14 ELBOW JOINT ASSEMBLY

1.	Collar	5.	Pulley
2.	Bearing housing	6.	link 2.60 x 40 x 3
3.	joint shaft	7.	link 3 taper channel
4.	Potentiometer		

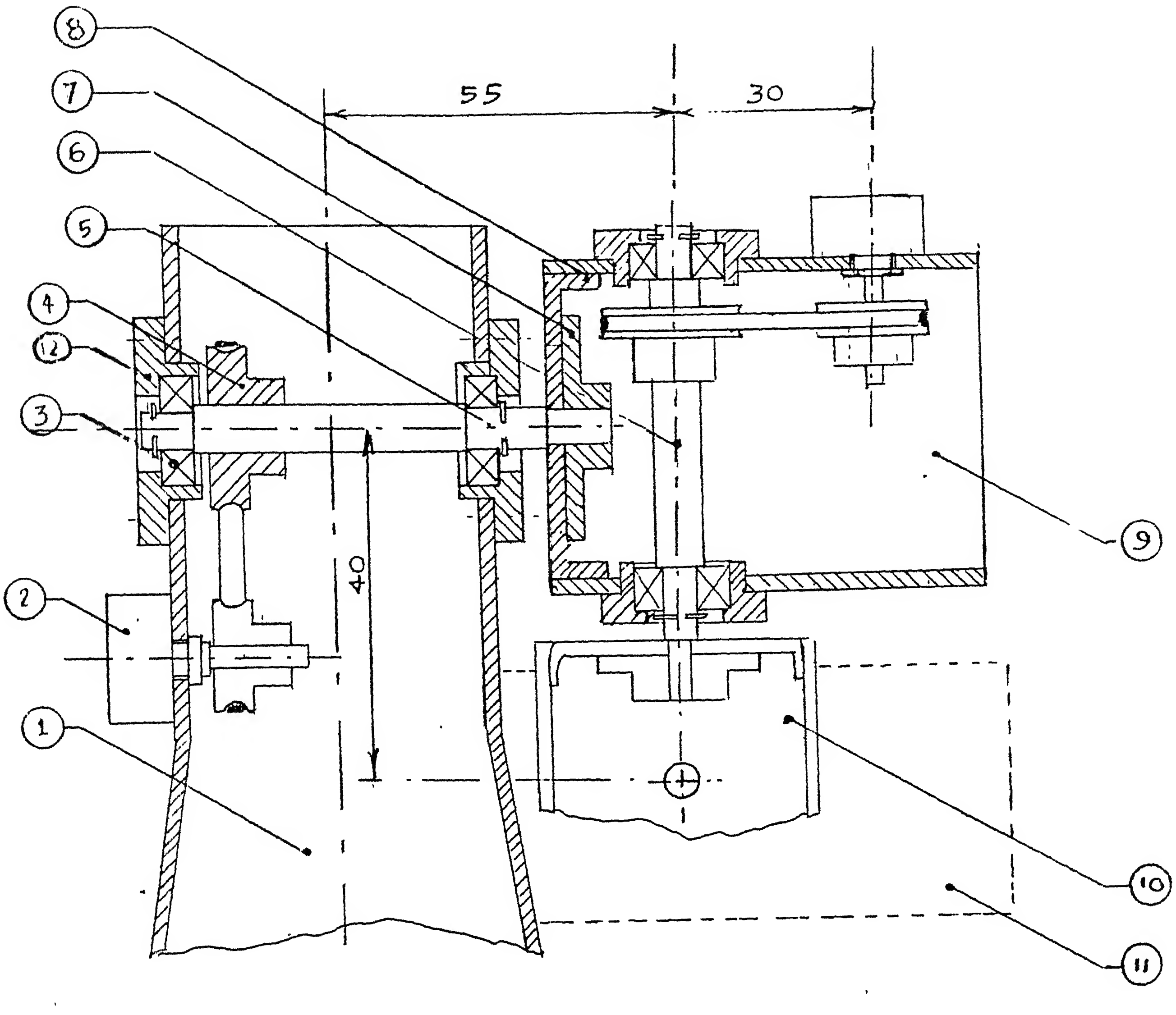


Fig.15. PITCH, YAW & ROLL JOINT ASSEMBLY

1.	link 3 taper channel	7.	Collar
2.	Potentiometer	8.	Support
3.	Bearing	9.	link 4 channel 32 x 30 x 3
4.	Pulley	10.	link 5 channel 40 x 30 x 3
5.	Joint 4 shaft	11.	Gripper (link 6)
6.	Joint 5 shaft	12.	Bearing housing

3.3 GRIPPER

Gripper and wrist design stressed on compactness, mechanical stiffness and dexterity. Gripper (Fig. 16)is made to accommodate left wrist and fingers of man. Thumb is fixed to the gripper to transfer the wrist motions while the fore fingers are inserted into a block hinging at one end. The distance between thumb and other fingers, which varies among people, can be adjusted to give effective working envelope for hand. The hinged block is spring loaded for return action and counter balancing. The gripper is also provided with a sliding switch to operate the slave gripper.

3.4 SENSORS AND TRANSMISSION

Potentiometers are chosen as position sensors in master arm due to their reliability in continuous operation and easy availability. Base and shoulder joints use three turn potentiometers and others single turn. They are driven from corresponding joint shafts by belt and pulley mechanism with suitable speed ratio such that the potentiometer range is fully utilized. For example a 270° potentiometer used to sense joint with 180° limits of motion is driven by pulley of speed ratio 3:2.

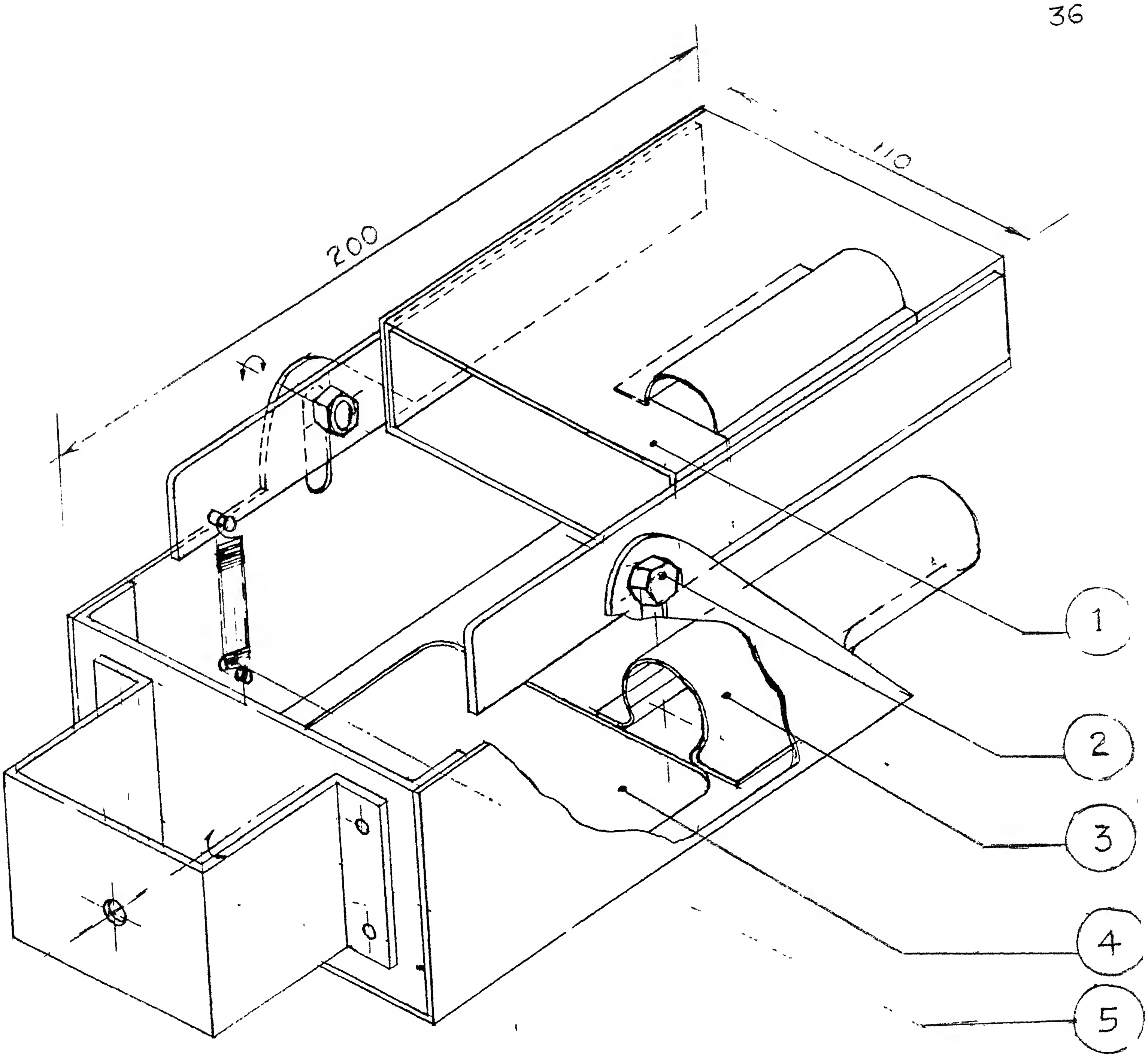


Fig.16 GRIPPER (MASTER ARM)

1. Hinged block for fingers
2. Adjustable screw
3. Thumb insert
4. Space for wrist insertion
5. Spring

3.5 MECHANICAL STOPS

The slave robot has limit switches at all joints and hence can execute only less than one revolution at joints. Mechanical stops are provided in the corresponding master arm joints so that all motions made by operator at master arm end are within permissible limits and hence executable at the slave end.

CHAPTER 4

CONTROL

4.1 PRINCIPLE

After analyzing the MA2000 robot system, it was found that it can be converted to a teleoperated robot by making changes such as freeing the robot from keypad interaction, elimination of lead by nose in continuous path mode and repetitive refreshing of memory to receive new data. In lead by nose- continuous path mode, during teaching the host computer gets position information at sampled intervals supplied by position sensors of the robot through the motor controller interface. The whole continuous path information is stored in host computer´s memory as an array starting from specific location and is treated as a single step while play back. The robot allows only one step of continuous path in the sequence of taught steps to be played back. While play back, these data points are transferred to the motor controller and the controller executes interpolated motion between sampled points so that the taught path is traced.

So, if master arm joint parameters after suitable calibration can be stored as set points in the computer memory and the assembly language routine executing interpolated motion can be called, the slave robot can be moved after a small delay, through the path traced by master arm. If this procedure is repeated infinitely updating joint parameters and calling routine for continuous path motion, the teleoperated robot can be realized.

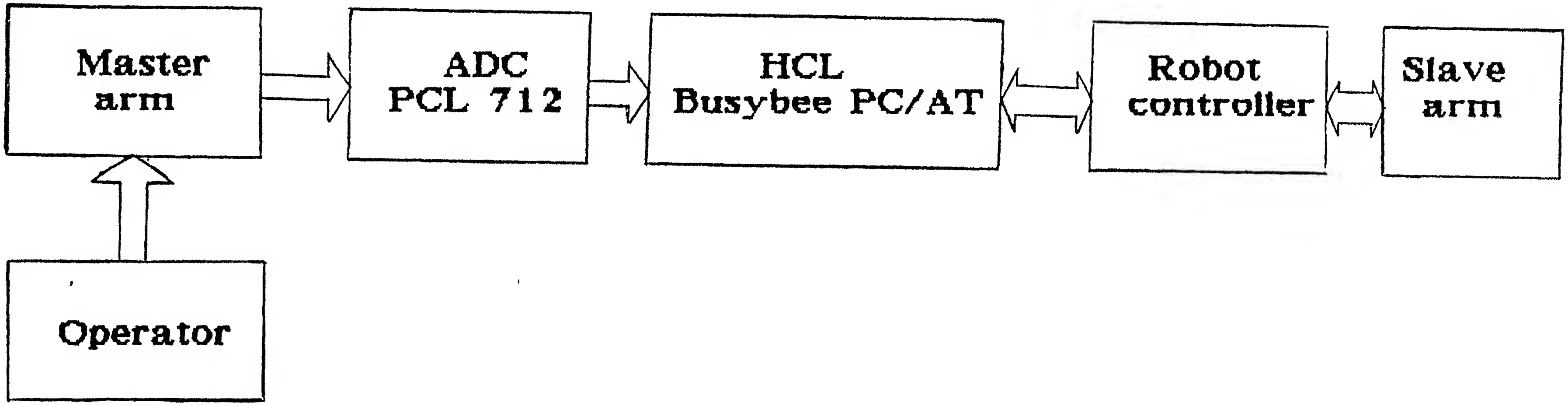
4.2 ELECTRONIC MODULES

A +5V supply is connected across the potentiometers of master arm. The intermediate points of the potentiometer are connected to a standard ADC card PCL712 [18]. Specifications of the card are given in appendix (2). The potential between ground and intermediate point of the joint potentiometer will indicate corresponding joint positions of the master. ADC card can be programmed to read input voltages. The MA2000-IBM interface card is used to communicate with the controller of the robot. The host computer used was HCL Busybee PC-AT (IBM compatible) in Robotics lab, IIT Kanpur. The master – slave interface is shown in Fig.17

4.3 CONTROL SOFTWARE:

An interactive and user friendly software was written in BBC BASIC to control the teleoperated robot. The source code listing is given in appendix (4) and instructions to use the system in appendix (3).

Fig 17. Master-Slave Interface



The software initialises by setting up hardware addresses and links between BASIC variables and assembly language routines. It then sets up 8255 into port A output, port B put and reads latch B to know the status of controller by sending read enable by setting bit 0 of C port to 1. If communication between controller and computer is perfect, it movcs the robot to park position which is the usual position of the master arm when not used. The movement is achieved by calling CALSISA and SAMOVE assembly language routines. Robot is now ready to act as teleoperator.

Calibration parameters are setup (ref.4.5) which converts the signals from potentiometers at master arm joints into corresponding slave joint parameters. ADC input channels are read and input voltage measured. They are converted to first set point. Like this, four set points are stored in memory locations accessible to assembly language routine CONTPATH which executes interpolated motion between these points. The memory refreshed and new sets of points are stored and CONTPATH is called again. This procedure is repeated infinitely. The flow chart for control software in shown Fig.18.

4.4. ERROR ELIMINATION

While testing the system, it was found that slave arm produced some chattering motion due to input errors.

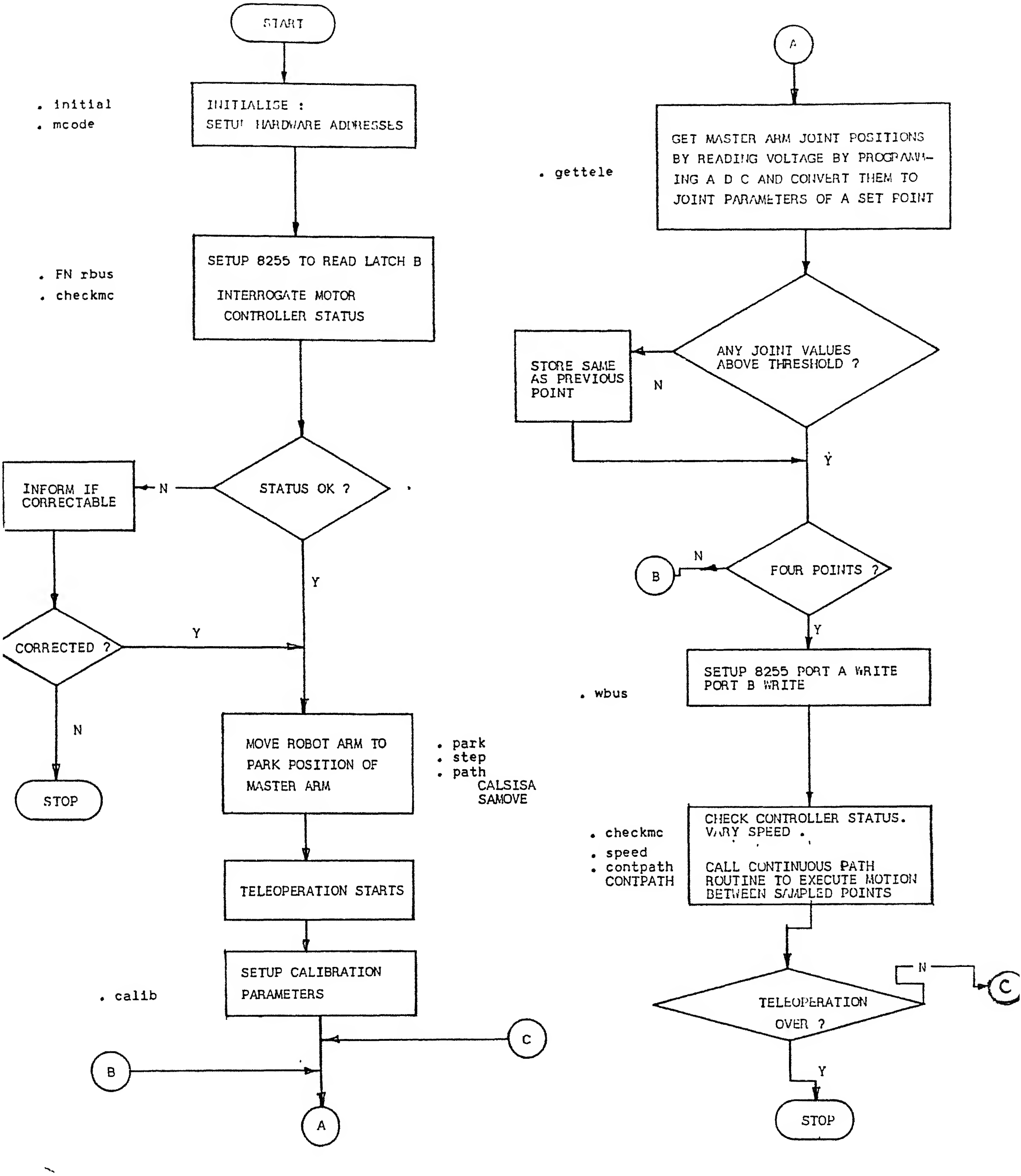


Fig 18. FLOW CHART OF CONTROL SOFTWARE

(Procedure names . by the side)

This was found to be of the order of +8 steps at each joint A threshold of ± 10 was setup to prevent chattering motion and was incorporated in the software.

4.5. CALIBRATION:

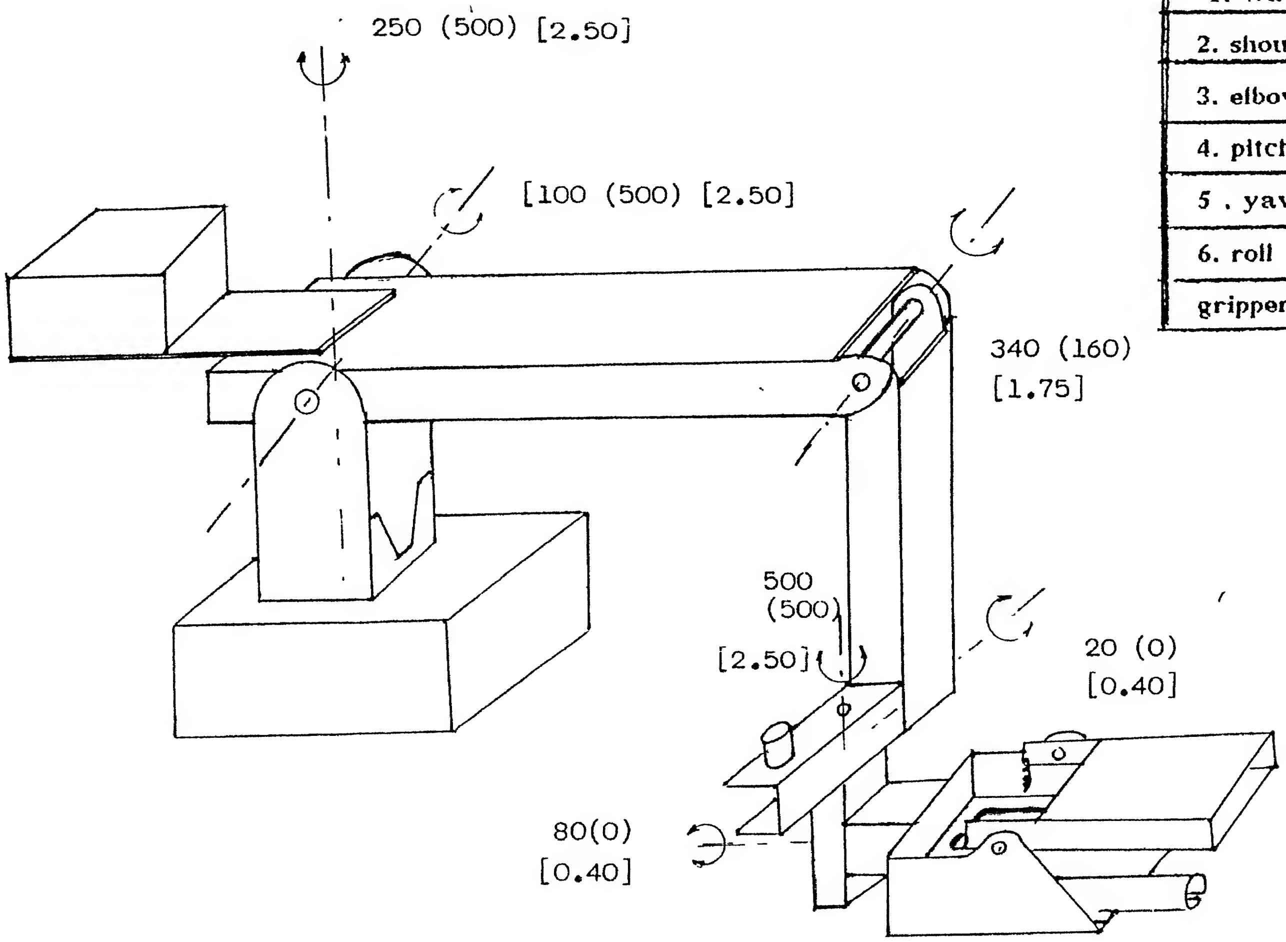
Each joint of slave robot can move in the range of microsteps from 0 - 999 in their limits of motion. Any joint position can be read as a step number by running the software supplied by thc manufacturer.

A reference configuration common to master and slave is chosen. (see Fig.19). With this, for each joint a reference position with respect to the previous joint was fixed and corresponding step number noted (No). Then the joint is rotated through a specific angle and the new step number noted (N). The difference $n = N - N_o$ found.

Each of the master arm joint is brought to the same reference position as the slave robot joint. After noting the value of the potentiometer at the joint (R_t), the potentiometer alone is turned so that it achieves a position permitting motion in the safe range, and resistance (Ro) between ground and intermediate point noted. The master arm joint after ensuring transmission of motion from joint shaft to potentiometer, is rotated through the same angle as

Fig.19. REFERENCE CONFIGURATION FOR CALIBRATION

Resistance (steps) [Voltages for ADC.C Program]



Joint	Pot. value (ohms)	Colour code	Channel in ADC
1. waist	497	red	1
2. shoulder	200	orange	7
3. elbow	970	yellow	3
4. pitch	1082	green	4
5 . yaw	1009	blue	5
6. roll	956	violet	6
gripper	-	ash	0
		white	+5 V
		black	ground

corresponding slave arm joint and new resistance reading noted (R_1) and the difference $r = R_1 - R_o$ is found. While turning the master arm joints care has been taken to ensure that the same sense of rotation produce similar variations in steps and resistance. This procedure is repeated for all the joints.

If a known input voltage is applied across the potentiometers (Vin) and at any intermediate position, ADC shows a voltage V at corresponding channel, then the joint step number (N) can be found from

$$N = No + \frac{n}{(r/\ Rt)\ Vin}\left[V - (Ro/\ Rt)\ Vin\right]$$

Later a program was developed to calibrate the system by bringing master arm to the reference position and noting the voltages directly (see appendix 3).

CHAPTER 5

PERFORMANCE OF THE SYSTEM

The master arm hardware was realized in IIT facilities. It was found to give human arm accommodation and manoeuvrability. Position sensors were connected to the joints and associated circuitry made. The standard ADC card used, initially gave multiplexing problems due to incorrect resistances used at the production level. The problem was eliminated by replacing with correct resistances. The control software developed was trial tested. The master slave interfaces made and the system was successfully tested with control software developed. The overall system as shown in Fig.20.

The software was found to be user friendly and it made the robot work independent of keypad which was previously essential for robot operations with manufacturer supplied software. The software gives good interaction by giving proper suggestions and warnings.

Tests were carried out with the operator manipulating objects by looking at video image of slave robot (Fig.21). Simple manipulations were easy; but slightly

Fig 20. Master- Slave Teleoperated Robot

complex operations were found to be difficult, though not impossible, mainly due to inability for depth perception in TV screen.

Tests were also performed to find response and accuracy of master slave system. The master arm gripper was given movement in x,y,z directions and corresponding movements of slave arm gripper was measured with respect to a parallel coordinate frame. Results were plotted as shown in Fig.22.

The error in movement of the slave arm are contributed by following factors:

- The master arm was designed to be kinematically equivalent to the slave robot; but the fabrication errors have disturbed this equivalence. Fabrication errors in link dimensions will produce accumulated error in end effector movement.

- There were errors in calibration since the instruments used to measure angle were not very sensitive. This can cause errors because of lack of correspondence in joint motions.

- The standard ADC used was found to give fluctuations in voltage probably due to noise in environment or imperfect

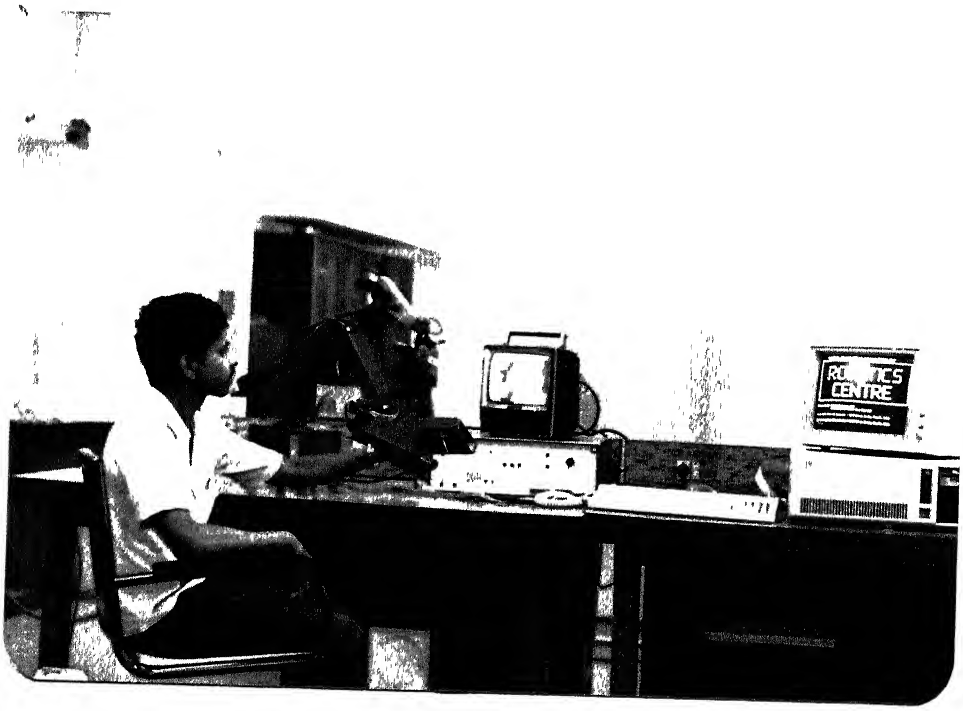


Fig 21. Teleoperated Robot- Tests with video.

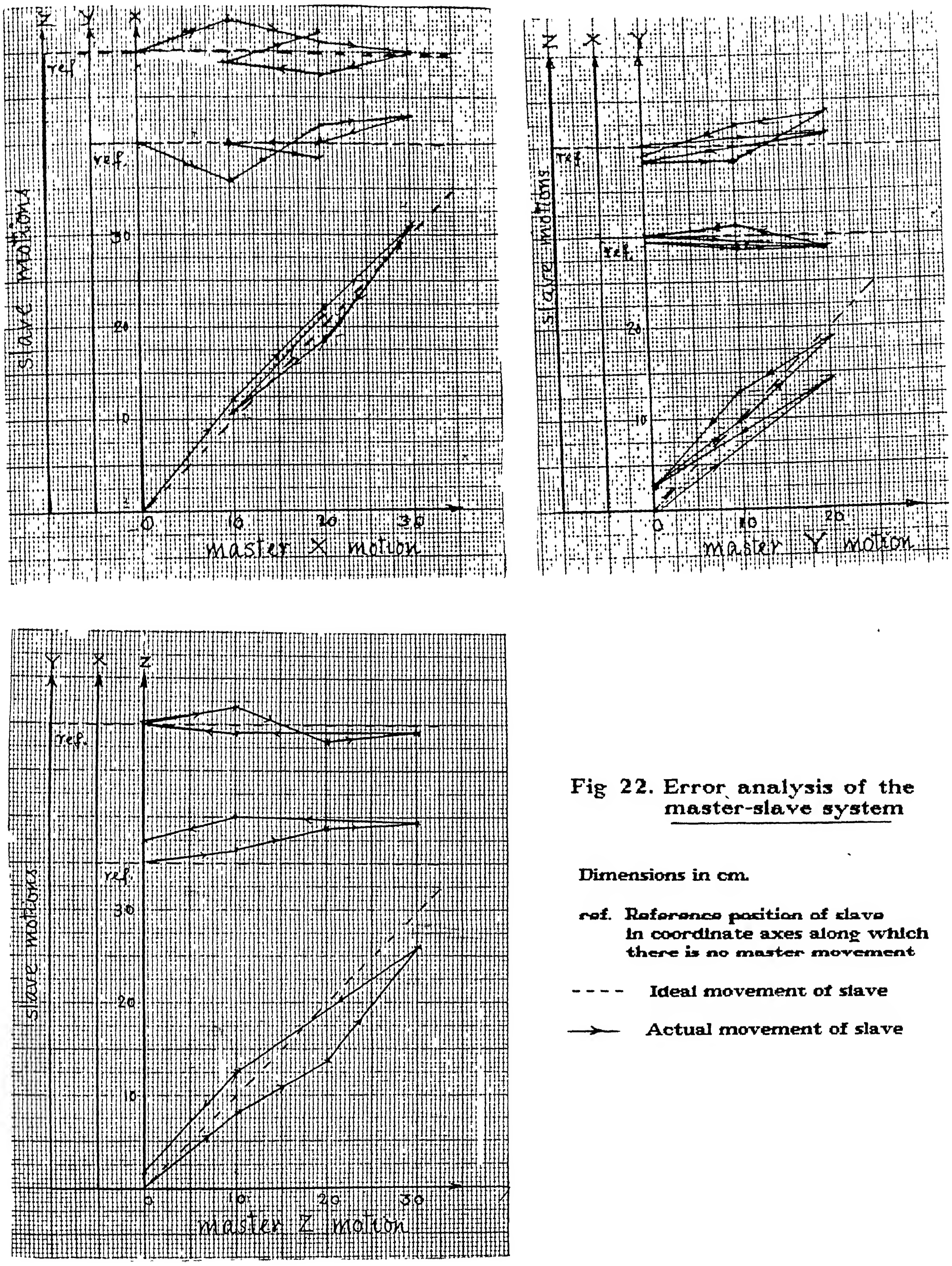


Fig 22. Error analysis of the master-slave system

Dimensions in cm.

ref. Reference position of slave in coordinate axes along which there is no master movement

- - - - Ideal movement of slave

—➤— Actual movement of slave

performance. They gave rise to errors of the order of ± 8 steps in every joint and a threshold was setup in software to eliminate it. This threshold can cause error in end effector position, by not being able to respond to micro movements.

- There were inaccuracies in setting up parallel coordinate frames. This caused measurement errors.
- The potentiometers used at master joint were not of very high quality. They can give rise to variable contact potential and non linearity over mechanical range or non linearity due to thermal variations.

Even with threshold setting in the software, the slave was seldom found to give erratic movements of approximately 15^{o} at the base and shoulder joints. Tests were carried out to detect the source of error.It was found that ADC gave rise to errors of ±10 steps and with potentiometer connected errors went upto ±40 steps. These can be reduced by filtering signals at the input to ADC and shielding of cables.

There was time delay in the response of the slave for motion executions. This can be attributed mainly to BBC Basic compilation, interpolation routines, and inherent delays in motor controller feedback loop and motor response. With the continuos path mode,it was not possible to control gripper with software.Later, a program was written using offline mode to get gripper control through software. Because of offline nature,response of the slave was slow and it moved in point to point manner. It can be used for pick and place operations.

Since visual feed back was used in deciding the course of action for the slave robot,these errors and delays did not cause much problem during manipulation.

CHAPTER 6

CONCLUSION

With the limitations in fabrication and instrumentation, the master-slave robot system performed well. With appropriate modifications it is hoped that it can meet the requirements of Indian nuclear, space and industrial sector and act as a stepping stone in the nascent area of ´telerobotics´. It can initiate variety of future projects.

6.1 SUGGESTIONS FOR FUTURE WORK

The errors in slave motion can be reduced to a considerable extent by using a in-house developed dedicated ADC, better positions sensors like optical encoders, and shielding the cables.

The error and sensitivity analysis of the teleoperators system is a problem to be studied and tackled for using teleoperator in complex and micro task environment.

The teleoperator can form part of a mobile inspection/assembly system by mounting the slave arm on a joystick controlled mobile platform having a teleoperated camera and lighting system. These mobile units will be more versatile than mobile Robot units available in foreign market due to teleoperated nature of arm. The mobile unit can be equipped with its own transmitter and receiver and thus the cable limiting the mobility can be discarded with. Such systems can be operated from remote locations.

The slave arm can be fitted with force sensors and master arm with actuators so that the operator gets force feedback and hence can perform complex operation. These systems can act as catalysts for works in the area of ´telepresence´.

Ergonomic studies can be carried out to study the operator fatigue and skill which will turn out data useful for many applications.

With better computational facilities and processors, the master arm can be replaced with a small joystick, which, though increases skill requirement of operator, will greatly reduce arm movements and the resultant fatigue.

PC compatibility of the developed master - slave system and use of robot as slave can open the new area of ´telerobotics´, which is the human supervision of semiautonomous systems. The human operator provides largely symbolic commands to the computer regarding goals, constraints, contingencies and assumptions while the subordinate telerobot executes the task based as information based on information received from the human operator plus its own artificial sensing and intelligence. The elimination/reduction of time delay in teleoperators loops is a research topic. Supervisory control was suggested as remedy, in which human operator interacts with the computer over the delayed channel and let the computer implement the commands unimpeded by the loop delay. Studies can be undertaken to implement supervisory control in the present system. As a beginning, during continuous path teleoperation, the robot can be stopped at a certain location and directed to execute a preprogrammed sequence of steps imparting some autonomy to the teleoperator.

BIBLIOGRAPHY

1. Bhabha Atomic Research Centre:
   Abstract of symposium on Advanced Remote handling systems and automation in nuclear installations : Bombay 1990

2. Davis P.R.; Stubbs DA : 'Safe levels of manual forces for young males' : Applied ergonomics: Sept 1977 (1); Dec 1977 (2)

3. Ferrel, Sheridan : 'Supervisory control of remote manipulation' : IEEE Spectrum Vol no 10, 1967 pp: 81-88

4. Francis N.Nagy, Ardras Seigler : 'Engg fundamentals of robotics' : Prentice Hall Inc, Englewood Cliffs 1986

5. International Encyclopedia of robotics : Vol III : John Wiley & Sons, NY 1988

6. Koji Ikuta, etal : 'Shape memory alloy servo actuator systems & application for active endoscope' 1988 IEEE conf. on robotics & Automation pp 427-433.

7. Lynn Conway, etal : "Tele autonomous systems: methods & architectures for intermingling autonomous & tele robotics technology' : Proc. IEEE conf. on robotics & automation 1987 : pp 1121-1130.

8. Martin H.L. : 'Joining teleoperation with robotics for remote manipulation in hostile environments ':Proc. Robots 8 conf: Detroit 1984 : 9-1 to 9-9.

9. Mc Kinnon, King : ´ Manual control of tele manipulation´ : proc. International symp. on teleoperators and control, Bristol 1988: PP: 263-276.

10. Sheridan T.B.: ´Telerobotics´ : Automatica, July 89 Vol.25 no.4 pp: 487-507.

11. Siva K.V, etal : ´Development of a general purpose hand controller for advanced teleoperation´: Proc. Intl symp on teleoperators and control : Bristol 1988 : pp 277-296.

12. Te Quipment Intl. Ltd. : MA2000 operator´s manual

13. Te Quipment Intl. Ltd. : MA2000 software & controller

14. Thring M.W. : ´Robots & Telechirs´ : Ellis horwood Ltd, UK 1983.

15. Todd. D.J. : ´Fundamentals of robot technology´ Kogan page, London 1986.

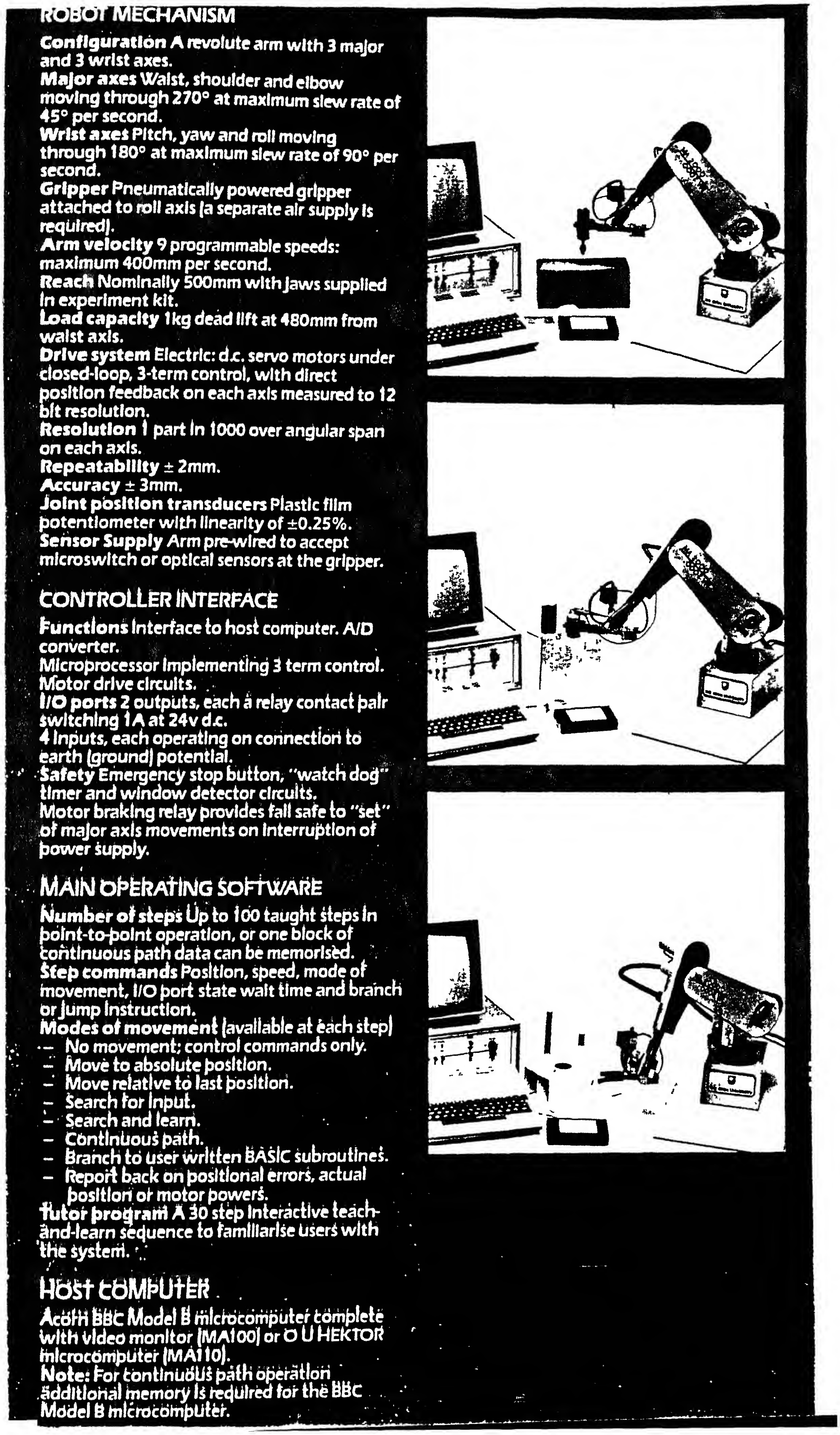
16. UK Atomic Energy authority : News item in Journal ´Atom´ : February 1990.

17. User´s guide : PCL Unicorn personal computer

18. User´s Manual : PCL 712 multilab card.

19. Vertut, Coiffet : ´Teleoperation and Robotics´ Vol.3A & 3B of Robot technology series: Kogan Page, London 1985.

Appendix 1. MA 2000 specifications



ROBOT MECHANISM

Configuration A revolute arm with 3 major and 3 wrist axes.
Major axes Waist, shoulder and elbow moving through 270° at maximum slew rate of 45° per second.
Wrist axes Pitch, yaw and roll moving through 180° at maximum slew rate of 90° per second.
Gripper Pneumatically powered gripper attached to roll axis (a separate air supply is required).
Arm velocity 9 programmable speeds: maximum 400mm per second.
Reach Nominally 500mm with jaws supplied in experiment kit.
Load capacity 1kg dead lift at 480mm from waist axis.
Drive system Electric: d.c. servo motors under closed-loop, 3-term control, with direct position feedback on each axis measured to 12 bit resolution.
Resolution 1 part in 1000 over angular span on each axis.
Repeatability ± 2mm.
Accuracy ± 3mm.
Joint position transducers Plastic film potentiometer with linearity of ±0.25%.
Sensor Supply Arm pre-wired to accept microswitch or optical sensors at the gripper.

CONTROLLER INTERFACE

Functions Interface to host computer. A/D converter.
Microprocessor implementing 3 term control.
Motor drive circuits.
I/O ports 2 outputs, each a relay contact pair switching 1A at 24v d.c.
4 inputs, each operating on connection to earth (ground) potential.
Safety Emergency stop button, "watch dog" timer and window detector circuits.
Motor braking relay provides fail safe to "set" of major axis movements on interruption of power supply.

MAIN OPERATING SOFTWARE

Number of steps Up to 100 taught steps in point-to-point operation, or one block of continuous path data can be memorised.
Step commands Position, speed, mode of movement, I/O port state wait time and branch or jump instruction.
Modes of movement (available at each step)

- No movement; control commands only.
- Move to absolute position.
- Move relative to last position.
- Search for input.
- Search and learn.
- Continuous path.
- Branch to user written BASIC subroutines.
- Report back on positional errors, actual position or motor powers.

Tutor program A 30 step interactive teach-and-learn sequence to familiarise users with the system.

HOST COMPUTER

Acorn BBC Model B microcomputer complete with video monitor (MA100) or O U HEKTOR microcomputer (MA110).
Note: For continuous path operation additional memory is required for the BBC Model B microcomputer.

Appendix 2

ADC PCL 712 SPECIFICATIONS

Analog to digital

Input Range	: Bipolar +/-5V
Input Channels	: 16 single ended
Accuracy	: +/-0.2% at +/-5V range +/-0.3% at +/-1V range
Input Impedance	: > 10 mega Ohms
Conversion time	: < 30 microseconds
Resolution	: 12 bits

Digital to analog

Output Range	: 0V to 5V
Output Channels	: 2
Accuracy	: 0.1%
Settling time	: < 30 microseconds for 5V step
Resolution	: 12 bits

Digital Input

Input Low Level	: Min. -0.5V, max. 0.8V
Input High Level	: Min. 2.0V, max. 5.0V
Input Loading	: 0.2 mA at 0.4V
Input Hysteresis	: Typical 0.4V, min. 0.2V

Digital Output

Output Low Level :	Max. 0.5V at 8 mA
(sink)	Max. 0.4V at 4 mA
Output High Level:	Min. 2.7V at 0.4 mA
(source)	

Power Consumption	: < 800 mA at 5V < 50 mA at +12V < 50 mA at -12V

General

Dimensions	: 13 3/8" x 3 3/4" 34 cm x 9.5 cm
Bus	: IBM PC bus
Slot	: One 62-pin slot
I/O Port Base Address	: Hex 200 - hex 3F0
I/O Port Space	: 12

The required switch settings for various base addresses are illustrated as below:

Note: - ON = 0 , OFF = 1
- "X" means "don't care"
- 1..5 are switch positions
- A4..A8 correspond to address lines of the PC bus. A9 is hard-wired to be 1.
- * means factory setting

I/O port address (Hex)	switch position 1 A8	2 A7	3 A6	4 A5	5 A4
200-20F	0	0	0	0	0
.					
.					
.					
*220-22F	0	0	0	1	0
.					
.					
.					
2E0-2EF	0	1	1	1	0
2F0-2FF	0	1	1	1	1
3E0-3EF	1	1	1	1	0
3F0-3FF	1	1	1	1	1

Connector Pin Assignments

The PCL-712 card is equipped with one 20-pin insulation displacement (mass termination) connector accessible from the rear plate and three other 20-pin insulation displacement connectors on board. All these connectors can be connected to flat cables of the same type. Please refer to Fig. 2.1 for the location of each connector.

The following diagrams below show their pin assignments.

Legend:

A/D	- single ended analog input
D/A	- analog output
D/O	- digital output
D/I	- digital input
GND	- ground
CLK	- the clock input for the 8253
GATE	- the gate input for the 8253
OUT	- the signal output of the 8253

Connector 1:

A/D 0	1	2	A/D 1
A/D 2	3	4	A/D 3
A/D 4	5	6	A/D 5
A/D 6	7	8	A/D 7
A/D 8	9	10	A/D 9
A/D 10	11	12	A/D 11
A/D 12	13	14	A/D 13
A/D 14	15	16	A/D 15
D/A 1	17	18	D/A 2
NOT USED	19	20	GND

Appendix 3

User's manual for master-slave teleoperated robot

1. Calibrate the master arm (ref Fig.19)

   a. ensure that

   - all motion transmission screws are tightened
   - all potentiometer contacts are correct
   - no belt slipping during motion

   b. Connect a +5V supply across the potentiometers

   c. Connect the joint potentiometers to corresponding channels.

   d. Run the turbo C program ADC.C to measure input voltages.

   e. Bring the master arm joints to reference position. Turn the potentiometer alone so that the voltage read ±0.06V of that given in square brackets.

NB: (i) If you have changed any of the potentiometers, proceed as follows

a: calibrate the system again as per 4.5

b: change variable, (Vj)i and (Pj)i in lines 370 to 420 of 'TELPATH' program

$(Pj)i = (N_o - n * R_o/r)i$

$(V_j)i = (n * R_t /(r * Vin)i$

where Vin input voltage while running the robot in teleoperator mode.

(ii) Default setting of Vin is + 5V. If you have to change it (but always ≤ 5V) change Vin in line no 370 of the program

2. Now Teleoperator is ready to function

```
CD\BBCBAS
BBCBASIC
> LOAD  "TELPATH"
> RUN
```

Ensure that controller is connected and is in AUTO mode. In case of any problem following massages will be shown

message	remedy
Controller status invalid	: Check that controller is 'on' and communication in good. Re run the program
Controller not in auto	Press <reset> <auto> of controller.

Press P to park the robot

type M for teleoperator mode

- press 'M' key

* now the slave is ready to work as teleoperator

* IN CASE OF ANY INVOLUNTARY MOVEMENT OF THE ROBOT PRESS EMERGENCY STOP IN THE CONTROLLER TO STOP THE ROBOT.

- If the involuntary motion is not serious press <reset> button to continue working as teleoperator

- If the movement seems to be hazardous escape from the program pressing <esc> in keyboard. THEN AND THEN ONLY, reset the controller.

NB: If the robot was driven out of limits adopt following procedure to bring within limit - Press the <reset> and <test> button together.

Keep <test> pressed and select the joint to be brought to limits and move it using front panel control.

APPENDIX 4

SOURCE CODE LISTING

```
 10 DATASTOPOSCLIOSCLIOSCLIOSCLI###
 20 HIMEM=&A000:AA%=&F600:FF%=PAGE
 30 I$="3A.05"
 40 REM COPYRIGHT (C) 1990 IIT KANPUR
 50  @%=2:VDU26:CLS:PRINT'"  TELEOPERATOR   PROJECT"'TAB(8)"IIT
    Kanpur"''TAB(12)' "Copyright (C)  1990 IIT KANPUR"''
 60 :
 70 REM Author: N.P.Giri, IIT Kanpur
 80 :
 90 N%=6
100 O%=TOP
110 PROCinitial:PROCmcode
120 PROCcheckseq
130 PROCcheckmc:IF stop VDU11,11,11:REPEAT PROCcheckmc:
    UNTIL NOTstop
140 FOR a%=0TO2:PROCout(a%,0):NEXT:PROCpark
150 IF CC%=0 ENDPROC ELSE mode%=6
160 PROCcalib
170 VP%=VPOS:I%=VP%+3
180 sa%(1)=500:sa%(2)=838:sa%(3)=160:sa%(4)=999:
    sa%(5)=500:sa%(6)=0:sa%(0)=0
190 FOR I%=1 TO 6:err%(I%)=sa%(I%):SPD%(I%)=sa%(I%):NEXT I%
200 PRINT:PRINT"Type 'M' for TELEOPERATOR mode"
210 REPEAT UNTIL GET$="M"
220 PRINT:PRINT"TELEOPERATOR IN ACTION":PRINT:
    PRINT"To STOP press <esc>"
230 REPEAT
240 PROCcheckmc
250 ptr%=CC%:J%=BB%-2*bytes%
260 FOR COUNT1= 1 TO 4:PROCgettele:PROCscale(1,6)
270 FORM%=1TON%:?ptr%=SA%(M%)MOD256:ptr%?1=SA%(M%)DIV256:
    ptr%=ptr%+2:NEXT
280 NEXT
290 PROCstatus(0):Z%=(ptr%-CC%)/(2*N%):PROCspeed:PROCcontpath
300 PROCgrip(sa%(0)):FOR I%=1 TO 6:SPD%(I%)=sa%(I%):NEXT I%
310 UNTIL FALSE
320 END
330 ===
340 :
350 DEF PROCcalib
360 REM ============
370 vin=5.0:pj1=338*0.498/(0.112*vin):vj1=500-338*0.25/0.11
380 pj2=338*0.201/(0.046*vin):vj2=500-338*0.1/0.046
390 pj3=340*0.97/(0.33*vin):vj3=160-340*0.34/0.33
400 pj4=500*1.082/(0.510*vin):vj4=500-500*0.6/0.510
410 pj5=300*0.956/(0.237*vin):vj5=500-300*0.5/0.237
420 pj6=500*1.009/(0.520*vin):vj6=-500*0.080/0.520
430 ENDPROC
```

```
440 DEF PROCspeed
450 REM =========
460 R%=1
470 REPEAT:mx%=ABS(sa%(R%)-SPD%(R%)):R%=R%+1:UNTIL R%>6 OR mx%>50
480 IF R%>6 OR (mx%>50 AND mx%<100) rate%=7:ENDPROC
490 IF (mx%>=100 AND mx%<150) rate%=8:ENDPROC
500 IF mx%>=150 rate%=9:ENDPROC
510 DEF PROCwbus(a%,d%)
520 REM ========
530 PUT   b%+3,136:PUT   b%,a%:PUT   b%+1,d%:PUT   b%+3,1:PUT
    b%+3,0:PUT b%+3,138:PUT b%+3,3:ENDPROC
540 :
550 DEF FNrbus(a%)
560 REM ======
570 LOCAL d%:PUT b%,a%:PUT b%+3,1:d%=GET(b%+1):PUT b%+3,0: = d%
580 :
590 DEF PROCpeek(s%)
600 REM ========
610 IF s%<0 ENDPROC ELSE ptr%=DD%+bytes%*s%
620 rate%=?ptr%:mode%=ptr%?1:input%=ptr%?2:output%=ptr%?3:
    wait%=ptr%?4:jump%=ptr%?5
630 IF jump%=255 jump%=-1
640 IF input%>127 input%=127-input%
650 IF output%>127 output%=127-output%
660 ptr%=ptr%+6:FORM%=0TON%:sa%(M%)=?ptr%+256*ptr%?1:
    ptr%=ptr%+2:NEXT
670 ENDPROC
680 :
690 DEF PROCpoke(s%)
700 REM ========
710 IF s%<0 ENDPROC ELSE ptr%=DD%+bytes%*s%
720 IF mode%<>11:IF input%<0 input%=127-input%
730 IF mode%<>11:IF output%<0 output%=127-output%
740 IF jump%<0 jump%=255
750 ?ptr%=rate%:ptr%?1=mode%:ptr%?2=input%:ptr%?3=output%:
    ptr%?4=wait%:ptr%?5=jump%
760 ptr%=ptr%+6:FORM%=0TON%:?ptr%=sa%(M%)MOD256:
    ptr%?1=sa%(M%)DIV256:ptr%=ptr%+2:NEXT
770 ENDPROC
780 :
790 DEF PROClimit(M%)
800 REM =========
810 LOCAL z%,h%:z%=0:h%=H%*2.001-1
820 IF M%=0 h%=1
830 IF M%=4ORM%=5 PROClimpy
840 IF sa%(M%)<z% sa%(M%)=z%:isa%(M%)=0:ENDPROC
850 IF sa%(M%)>h% sa%(M%)=h%:isa%(M%)=0:ENDPROC
860 ENDPROC
870 DEF PROClimpy
880 IF sa%(5)>450 ENDPROC
890 IF sa%(4)<200:IF sa%(5)-sa%(4)<250  sa%(M%)=sa%(9-M%)+250:
    isa%(M%)=0:ENDPROC
```

```
 900 IF sa%(4)>=200:IF sa%(5)+sa%(4)<650  sa%(M%)=650-sa%(9-M%):
     lsa%(M%)=0:ENDPROC
 910 ENDPROC
 920 :
 930 DEF PROCscale(m%,n%)
 940 REM =========
 950 LOCAL i%:FORi%=m%TOn%
 960 LSA%(i%)=lsa%(i%)*span(i%)+intcpt%(i%):
     SA%(i%)=sa%(i%)*span(i%)+intcpt%(i%)
 970 NEXT:ENDPROC
 980 :
 990 DEF PROCdescale
1000 REM ===========
1010 FORM%=1TON%:sa%(M%)=(SA%(M%)-intcpt%(M%))/span(M%)+0.4:NEXT
1020 ENDPROC
1030 :
1040 DEF FNverify
1050 REM ========
1060 IF S%<0 OR S%>E% = FALSE
1070 IF mode%=10ANDNOTl% THEN=FALSE ELSE IF mode%=1 OR  mode%=11
     OR mode%>5 = TRUE
1080 J=0:FORM%=1TON%:J=J+sa%(M%):NEXT
1090 IF rate%>0 AND rate%<10 AND mode%>0 AND mode%<7  AND  J>=0
     AND J<=N%*H%*2 THEN = TRUE ELSE = FALSE
1100 :
1110 :
1120 DEF PROCstep
1130 REM ========
1140 PROCwatchdog
1150 good=FNverify:IF NOTgood PROCfault:ENDPROC
1160 IF  stop  COLOURfl%:PRINT''"  EMERGENCY  STOP   pressed"':
     COLOURwh%:VDU7,7,7,7,7,7,7,7: ENDPROC
1170 IFmode%>1 AND mode%<6 PROCpath
1180 lastS%=S%:S%=S%+1:PROCpeek(S%)
1190 ENDPROC
1200 :
1210 DEF PROCpath
1220 REM ========
1230 LOCAL a%,in%,s%
1240 PROCscale(1,N%):LSA%(0)=lsa%(0)
1250 CALL CALCISA,sa%(1),lsa%(1),isa%(1),WSPA%(0)
1260 PROCwbus(keypad,(outa%AND224)+16):hkey%=0
1270 IF rate%>4:IF WSPA%(0)>60 THEN PROCstatus(3)
1280 CALL SAMOVE,rate%,mode%,hkey%,ipmska%,ipmskb%,ipres%,
     outb%,SA%(0),LSA%(0),isa%(1):PROCstatus(0)
1290 PROCgrip(sa%(0))
1300 FORM%=0TON%:lsa%(M%)=sa%(M%):NEXT:lastS%=S%
1310 ENDPROC
1320 :
1330 DEF PROCout(a%,out%)
1340 REM =======
1350 LOCAL o%,v%,j%:j%=ABS(out%)
1360 o%=0:REPEATo%=o%+2^(j%MOD10-1):j%=j%DIV10:UNTILj%=0
```

```
1370 IF out%<0 THEN o%=o% EOR 255:out%(a%)=out%(a%) AND o%  ELSE
     out%(a%)=out%(a%) OR o%
1380 IF a%=0 outb%=(outb%AND240)+out%(a%):
     PROCwbus(procout%(a%),outb%):ENDPROC
1390 PROCwbus(procout%(a%),out%(a%)):ENDPROC
1400 :
1410 DEF PROCgrip(g)
1420 REM ========
1430 outa%=128*g+(outa%AND127):PROCwbus(gripout,outa%):ENDPROC
1440 :
1450 DEF PROCwatchdog
1460 REM ============
1470 PROCwbus(&7F,0):ENDPROC
1480 :
1490 DEF PROCstatus(state)
1500 REM ==========
1510 IF state>0 state=state+3
1520 outb%=(outb% AND15)+state*16:PROCwbus(portb,outb%):ENDPROC
1530 :
1540 DEF FNyes
1550 REM =====
1560 *FX21,0
1570 IF GET$="N"THEN PRINT"NO"':=FALSE ELSE PRINT"YES"':=TRUE
1580 :
1590 DEF PROCwhere
1600 REM =========
1610 FORI%=1TO5:PROCstatus(2):CALL REPORTEPP,outb%,WSPA%(1),
     WSPB%(1),WSPC%(1):PROCstatus(0):NEXT
1620 FOR M%=1TON%:SA%(M%)=WSPB%(M%):NEXT:PROCdescale:
     FOR M%=1TON%:PROClimit(M%):lsa%(M%)=sa%(M%):NEXT
1630 ENDPROC
1640 :
1650 DEF PROCclearall
1660 REM ============
1670 LOCAL I,J
1680 PRINT"Clearing DATA AREA, please wait."'
1690 I=DD%-&10:E%=0:?I=85:I?1=170:I?2=bytes%:
     I?3=T%:I!4=0:I!8=0:I!12=0
1700 FOR S%=1TOTS%:ptr%=DD%+bytes%*S%:PRINTS%:VDU11:
     FOR I%=0TObytes%:ptr%?I%=0:NEXT:NEXT:VDU11:S%=0
1710 ENDPROC
1720 :
1730 DEF PROCpark
1740 REM ========
1750 PROCwhere
1760 rate%=7:mode%=2:input%=0:output%=0:wait%=0:jump%=0
1770 PROCgrip(0):REPEAT PROCcheckmc:UNTIL NOTstop
1780 T=0:CLS:PRINT:PROCi:PRINT"Check that it is";:PROCn:PRINT''"  ";:
     PROCi:PRINT"SAFE to MOVE the ROBOT";:PROCn:PRINT:PRINT
1790 PRINT'"First PARK the ROBOT. Press P"'
1800 REPEAT:UNTIL GET$="P"
1810 REPEAT PROCcheckmc:UNTIL NOTstop:CLS
1820 PRINT'"Moving to PARK position"'
```

```
1830 sa%(1)=500:sa%(2)=838:sa%(3)=160:sa%(4)=999:sa%(5)=500:
     sa%(6)=0:sa%(0)=0
1840 PROCpoke(0):PROCstep:ENDPROC
1850 :
1860 DEF PROCcheckseq
1870 REM ============
1880 DD%=(HIMEM+&30)AND&FFF0:BB%=AA%-bytes%-1:
     TS%=(BB%-DD%)/bytes%:IF TS%>250 TS%=250
1890 IF T%>TS% OR T%=0 THEN T%=TS% ELSE TS%=T%
1900 IF c% CC%=DD%+(T%+2)*bytes%:IF CC%>BB%-N%*2 c%=FALSE
1910 I=DD%-&10:IF ?I=85 AND I?1=170 AND I?2=bytes% THEN T%=I?3:
     E%=I?4:seq=I?5+256*(I?6):Z%=I!7:tool%=I?11:ENDPROC
1920 ?I=85:I?1=170:IF ?I<>85 OR I?1<>170 COLOURfl%:PRINT'
     "CAN NOT ACCESS STORED-SEQUENCE MEMORY"':COLOURwh%:PROCfatal
1930 PRINT'"Clear data area (Y/N) ";:IF FNyes THEN PROCclearall:
     ENDPROC
1940 PRINT"Are you SURE";:PRINT''"that data area is NOT
     corrupted?!"'"(Y/N) ";:IF FNyes ENDPROC ELSE PROCclearall
1950 ENDPROC
1960 :
1970 DEF PROCcheckmc
1980 REM ===========
1990 LOCAL k
2000 k=FNrbus(portb)AND 6
2010 IF k=6 OR ((k=0)ANDgood) COLOURfl%:PRINT'" CONTROLLER
     STATUS INVALID":COLOURwh%:PROCfatal:ENDPROC
2020 IF k <>2 COLOURfl%:PRINT'" CONTROLLER NOT IN AUTO":
     VDU7,7,7,7,7:REPEAT PROCwatchdog:k=(FNrbus(portb))AND 2:
     UNTIL k:VDU13,11:PRINTTAB(100):VDU13,11,11,11:PROCwhere
2030 k=(FNrbus(portb))AND8:IF k AND NOTstop THEN stop=TRUE:
     COLOURfl%:PRINT''" EMERGENCY STOP pressed":COLOURwh%:
     VDU7,7,7,7,7,11:ENDPROC ELSE IF k=0 AND stop THEN stop=
     FALSE:PRINT" EMERGENCY STOP released"':PROCwhere
2040 ENDPROC
2050 :
2060 DEF PROCfatal
2070 REM =========
2080 *FX 214,254
2090 COLOURfl%:PRINT''TAB(5)"!!! FATAL ERROR !!!"':PROCn
2100 VDU7:TIME=0:REPEAT UNTILTIME>6000:RUN
2110 ENDPROC
2120 :
2130 DEF PROCptrs
2140 REM ========
2150 I%=DD%-&10:I%?3=T%:I%?4=E%:I%?5=seq MOD256:
     I%?6=seq DIV256:I%!7=Z%:I%?11=tool%
2160 ENDPROC
2170 :
2180 DEF PROCset
2190 REM =======
2200 FORM%=0TON%:isa%(M%)=0:NEXT:ENDPROC
2210 :
```

```
2220 DEF PROCinitial
2230 REM ===========
2240 b%=&300:REM Base Address of 8255 interface
2250 PUT b%+3,138:PUT b%+3,3:REM initialise interface
2260 fl%=31:wh%=7:REM screen colours
2270 *FX15
2280 T%=FNflag(0):l%=(INT(VAL(I$))=5)
2290 c%=FNflag(1):o%=FNflag(2):x%=FALSE:y%=FALSE:t%=FNflag(5):
     u%=FNflag(6)
2300 drive=FALSE:run=FALSE:edit=FALSE:stop=FALSE:good=FALSE
2310 S%=0:lastS%=0:out%=0:outa%=0:outb%=0:bytes%=(N%+1)*2+6
2320 LK%=0:seq=0:seq$="":H%=500:ptr=0:O$="P"
2330 kdisp%=0:wai%=100:xyz%=0
2340 DIM span(6),intcpt%(6)
2350 RESTORE 2530:READ I$:IF I$>"" PRINT"Serial No. "I$
2360 FORM%=1TON%:READ span(M%):NEXT:FORM%=1TON%:
     READ intcpt%(M%):NEXT
2370 DIM isa%(7),sa%(15),lsa%(7),procin%(2),procout%(2),
     out%(3),err%(15),SPD%(15)
2380 access$="     ":code99$="0184"
2390 :
2400 REM MA-Interface addresses
2410 setptl=0:setpth=&8:err=&10:posl=&20:posh=&28:pwr=&30
2420 porta=&81:portb=&80
2430 pgain=&38:igain=&40:dgain=&48
2440 keypad=porta:gripout=porta
2450 procin%(0)=porta:procout%(0)=portb
2460 procin%(1)=&82:procout%(1)=&82:procin%(2)=&83:procout%(2)=&83
2470 FORI%=0TO3:out%(I%)=0:NEXT
2480 :
2490 DIM kbmd% 12,kbid% 12,kpn% 20
2500 ?kbmd%=0:kbmd%!1=&04030201:kbmd%!5=&05040605:kbmd%!9=&03020106
2510 ?kbid%=0:kbid%!1=&00020202:kbid%!5=&02020000:kbid%!9=&00000002
2520 kpn%!1=&0704010A:kpn%!5=&0805020B:kpn%!9=&0906030C:
     kpn%!13=&100F0E0D:kpn%!17=&14121113
2530 :
2540 DATA ""
2550 DATA 3.25,3.25,3.25,2.80,2.80,2.80
2560 DATA  423, 423, 423, 648, 648, 648
2570 ENDPROC
2580 :
2590 DEF FNflag(I%)
2600 REM ======
2610 IF I% THEN =(FF%?(I%+4)=255) ELSE =FF%?4
2620 :
2630 DEF PROCi:COLOUR0:COLOUR128+wh%:PRINT" ";:ENDPROC
2640 REM =====
2650 DEF PROCn:PRINT" ";:COLOURwh%:COLOUR128:ENDPROC
2660 REM =====
2670 :
2680 DEF PROCmcode
2690 REM =========
2700 DIM WSPA%(7),WSPB%(7),WSPC%(7),SA%(7),LSA%(7)
```

```
2710 ipadd%=0:ipmska%=0:ipmskb%=0:ipres%=0:ptr%=0
2720 lbyte%=0:hbyte%=0:drivem%=0:hkey%=0
2730 REM
2740 OSCLI"LOAD MA2-"+STR$(N%)+" "+STR$~(AA%)
2750 SAMOVE=AA%+&95:SETLIMIT=AA%+&2C1:DRIVESA=AA%+&2DA:
     LEADSA=AA%+&3A8:REPORTEPP=AA%+&3FF:REPORTPID=AA%+&496:
     SETPID=AA%+&51D:KEYPAD=AA%+&5AC:CALCISA=AA%+&625:
     CONTPATH=AA%+&6AC
2760 FOR M%=1TON%:WSPA%(M%)=INT((H%*2.001-1))*span(M%)+intcpt%(M%):
     WSPB%(M%)=intcpt%(M%):NEXT
2770 CALL SETLIMIT,WSPA%(1),WSPB%(1)
2780 rate%=0:mode%=0:FORI=0TO7:WSPA%(I)=0:WSPB%(I)=0:
     WSPC%(I)=0:NEXT
2790 ENDPROC
2800 :
2810 DEF PROCcontpath
2820 REM ============
2830 PROCscale(1,N%):PROCwbus(keypad,(outa%AND224)+16):
     hkey%=0:PROCstatus(3)
2840 WSPC%(1)=CC%:WSPC%(2)=CC%+Z%*2*N%
2850 CALL CONTPATH,rate%,mode%,hkey%,outb%,LSA%(0),WSPC%(1)
2860 PROCstatus(0):lastS%=S%:PROCdescale:FORM%=0TON%:
     lsa%(M%)=sa%(M%):NEXT
2870 ENDPROC
2880 :
2890 DEF PROCgettele
2900 REM ===============
2910 FOR CH%=0 TO 7
2920 BASE%=&3E0
2930 PUT BASE%+10,CH%
2940 PUT BASE%+11,1
2950 HI=GET(BASE%+5)
2960 IF HI>=16 THEN GOTO 2950
2970 LO=GET(BASE%+4)
2980 V=(HI*256+LO-2048)*10/4096
2990 IF CH%=1 sa%(1)=vj1+pj1*V
3000 IF CH%=7 sa%(2)=vj2+pj2*V
3010 IF CH%=3 sa%(3)=vj3+pj3*V
3020 IF CH%=0 AND V<1 sa%(0)=0
3030 IF CH%=0 AND V>1 sa%(0)=1
3040 IF CH%=4 sa%(4)=vj4+pj4*V
3050 IF CH%=5 sa%(5)=vj5+pj5*V
3060 IF sa%(5)<250 sa%(5)=260
3070 IF sa%(5)>750 sa%(5)=750
3080 IF CH%=6 sa%(6)=vj6+pj6*V
3090 NEXT
3100 FOR M%=0 TO N%:PROClimit(M%):NEXT
3110 Q%=0
3120 REPEAT
3130 Q%=Q%+1
3140 UNTIL Q%>6 OR (ABS(sa%(Q%)-err%(Q%)) >30)
3150 IF Q%>6 GOTO 3170
3160 FOR Q%=1 TO 6:err%(Q%)=sa%(Q%):NEXT:ENDPROC
3170 FOR Q%=1 TO 6:sa%(Q%)=err%(Q%):NEXT:ENDPROC
```